

Beatriz Pama de Vasconcelos

Isabelle Ritter Vargas

Yuri de Sene Alvizi

Análise comparativa de algoritmos aplicados à análise da concessão de crédito bancário

São Paulo, SP

2025

Beatriz Pama de Vasconcelos

Isabelle Ritter Vargas

Yuri de Sene Alvizi

Análise comparativa de algoritmos aplicados à análise da concessão de crédito bancário

Trabalho de conclusão de curso apresentado
ao Departamento de Engenharia de Computação e Sistemas Digitais da Escola Politécnica
da Universidade de São Paulo para obtenção
do Título de Engenheiro.

Universidade de São Paulo – USP

Escola Politécnica

Departamento de Engenharia de Computação e Sistemas Digitais (PCS)

Orientador: Prof. Dr. Reginaldo Arakaki

São Paulo, SP

2025

Autorizo a reprodução e divulgação total ou parcial deste trabalho, por qualquer meio convencional ou eletrônico, para fins de estudo e pesquisa, desde que citada a fonte.

Catalogação-na-publicação

Vasconcelos, Beatriz

Análise comparativa de algoritmos aplicados à análise da concessão de crédito bancário / B. Vasconcelos, I. Vargas, Y. Alvizi -- São Paulo, 2025.
83 p.

Trabalho de Formatura - Escola Politécnica da Universidade de São Paulo. Departamento de Engenharia de Computação e Sistemas Digitais.

1.Concessão de crédito 2.Machine Learning 3.Análise de trade-offs
4.ATAM 5.Arquitetura de software I.Universidade de São Paulo. Escola Politécnica. Departamento de Engenharia de Computação e Sistemas Digitais
II.t. III.Vargas, Isabelle IV.Alvizi, Yuri

Agradecimentos

Agradecemos, em primeiro lugar, ao nosso orientador, Prof. Dr. Reginaldo Arakaki, pela orientação, pelas contribuições valiosas e pela dedicação ao longo de todo o desenvolvimento deste trabalho. Estendemos nossos agradecimentos a todos os professores da graduação, que, com seu conhecimento e empenho, contribuíram de forma essencial para nossa formação acadêmica e pessoal. Agradecemos também aos nossos amigos e a todas as pessoas que, de alguma maneira, colaboraram com este projeto — seja com palavras de incentivo, apoio técnico ou companhia nos momentos desafiadores.

Beatriz Pama de Vasconcelos

Agradeço, primeiramente, à minha avó Meiri, minha maior fonte de inspiração. Sua força, resiliência e coragem sempre foram exemplos que me motivaram a seguir em frente. Aos meus pais, Rita e Luiz, por todo amor, apoio e confiança, e por nunca medirem esforços para que eu pudesse realizar meus sonhos e ser feliz. Ao meu irmão, Lucas, por me inspirar a ser sempre melhor e servir de exemplo em tantos aspectos. E, por fim, às minhas amigas, pela escuta atenta, pela companhia constante e pelo apoio incondicional ao longo de todos esses anos de amizade.

Isabelle Ritter Vargas

A Deus, por iluminar cada um dos meus passos e me conceder as ferramentas necessárias para superar todas as dificuldades ao longo dessa jornada. Aos meus pais, Rose e Adão, por acreditarem em mim mais do que eu mesma e por nunca medirem esforços para me apoiar. O exemplo de dedicação, honestidade e trabalho árduo de vocês me inspira diariamente e me guia em todas as escolhas. Ao meu irmão, Felipe, por ser um pedaço do meu coração fora do peito e por sempre estar ao meu lado. E, por fim, ao meu namorado, Guilherme, que vibrou comigo em cada conquista e sempre foi meu maior ouvinte e apoiador.

Yuri de Sene Alvizi

Gostaria de agradecer, primeiramente, aos meus pais, Sirlene e Élcio, que desde cedo me ensinaram o valor dos estudos e sempre fizeram o possível para garantir minha felicidade e oferecer as melhores condições para meu crescimento. O apoio incondicional deles foi essencial em toda a minha trajetória. Agradeço também à minha namorada, Isabela, pela sua presença, incentivo e confiança no meu potencial. Estendo meus agradecimentos à minha família, amigos e professores, que, cada um à sua maneira, contribuíram para meu desenvolvimento e para todas as conquistas que alcancei até aqui.

Resumo

Este trabalho aborda o desafio da seleção de modelos de *machine learning* para a concessão de crédito bancário, propondo um método estruturado que alinhe o desempenho técnico às necessidades do negócio. O diferencial metodológico reside na aplicação do *Architecture Tradeoff Analysis Method* (ATAM) como um framework para avaliar sistematicamente os *trade-offs* entre atributos de qualidade essenciais, como a performance preditiva e o tempo de treinamento. Para validar essa abordagem, diferentes classes de algoritmos, incluindo modelos lineares, *ensembles* baseados em árvores e redes neurais, foram implementadas como arquiteturas candidatas e treinadas em um conjunto de dados expandido sinteticamente. A aplicação do método ATAM permitiu identificar a arquitetura que ofereceu o balanço mais favorável entre os requisitos conflitantes, como a obtenção de um alto poder preditivo com um tempo de treinamento adequado ao negócio, resultando em uma recomendação fundamentada. Conclui-se que a adoção de um *framework* de análise formal é fundamental para transcender a simples comparação de métricas, permitindo uma tomada de decisão que integra as capacidades técnicas dos algoritmos às prioridades estratégicas e operacionais da instituição financeira.

Palavras-chave: Concessão de crédito. *Machine learning*. Análise de *trade-offs*. ATAM. Arquitetura de software.

Abstract

This work addresses the challenge of selecting machine learning models for bank credit granting by proposing a structured method that aligns technical performance with business needs. The methodological differential lies in the application of the Architecture Tradeoff Analysis Method (ATAM) as a framework to systematically evaluate the trade-offs between essential quality attributes, such as predictive performance and training time. To validate this approach, different classes of algorithms, including linear models, tree-based ensembles, and neural networks, were implemented as candidate architectures and trained on a synthetically expanded dataset. The application of the ATAM method made it possible to identify the architecture that offered the most favorable balance among conflicting requirements, such as achieving high predictive power with a business-appropriate training time, resulting in a well-founded recommendation. It is concluded that the adoption of a formal analysis framework is fundamental to transcend a mere comparison of metrics, enabling a decision-making process that integrates the technical capabilities of the algorithms with the strategic and operational priorities of the financial institution.

Keywords: Credit granting. Machine learning. Trade-off analysis. ATAM. Software architecture.

Listas de ilustrações

Figura 1 – Tipos de riscos em bancos	14
Figura 2 – Composição da receita em bancos	15
Figura 3 – Inadimplência	15
Figura 4 – Diagrama de uma Rede Neural	18
Figura 5 – Modelo matemático de um neurônio	18
Figura 6 – Funcionamento de uma GAN	19
Figura 7 – Esquema do Random Forest	21
Figura 8 – Visões Arquiteturais	29
Figura 9 – BPMN	31
Figura 10 – Fluxo de Processos de Engenharia	34
Figura 11 – YData	35
Figura 12 – Scikit Learn	36
Figura 13 – Distribuição de risco por grupo de idade	39
Figura 14 – Distribuição de risco por sexo	39
Figura 15 – Comportamento de mau pagadores em diferentes faixas de crédito	40
Figura 16 – Distribuição de risco por faixa de poupança	41
Figura 17 – Distribuição de risco por tempo de permanência no emprego atual	42
Figura 18 – Métricas	44
Figura 19 – Regressão Logística - Matriz de confusão	45
Figura 20 – Random Forest - Matriz de confusão	47
Figura 21 – XGBoost - Matriz de confusão	49
Figura 22 – MLP - Matriz de confusão	51
Figura 23 – Métricas dos algoritmos	54
Figura 24 – Importância global das variáveis segundo o modelo Regressão Logística.	67
Figura 25 – Summary plot do modelo Regressão Logística.	68
Figura 26 – Waterfall plot do modelo Regressão Logística para a amostra selecionada.	69
Figura 27 – Importância global das variáveis segundo o modelo <i>Random Forest</i>	70
Figura 28 – Summary plot do modelo <i>Random Forest</i>	71
Figura 29 – Waterfall plot do modelo <i>Random Forest</i> para a amostra selecionada.	71
Figura 30 – Importância global das variáveis segundo o modelo XGBoost.	72
Figura 31 – Summary plot do modelo XGBoost.	73
Figura 32 – Waterfall plot do modelo XGBoost para a amostra selecionada.	74
Figura 33 – Importância global das variáveis segundo o modelo MLP.	74
Figura 34 – Summary plot do modelo MLP.	75
Figura 35 – Waterfall plot do modelo MLP para a amostra selecionada.	76

Lista de tabelas

Tabela 1 – Colunas do dataset	37
Tabela 2 – Matriz de Confusão	42
Tabela 3 – Parâmetros utilizados	44
Tabela 4 – Regressão Logística - Resultados obtidos	45
Tabela 5 – Random Forest - Resultados obtidos	46
Tabela 6 – XGBoost - Parâmetros utilizados	48
Tabela 7 – XGBoost - Resultados obtidos	49
Tabela 8 – MLP - Parâmetros utilizados	50
Tabela 9 – MLP - Resultados obtidos	50
Tabela 10 – Resultados obtidos	53
Tabela 11 – Perfil do cliente utilizado para comparação dos waterfall plots nos quatro modelos.	66

Sumário

1	INTRODUÇÃO	11
1.1	Motivação	11
1.1.1	Dinâmicas do mercado	12
1.1.2	Perfil dos consumidores e do mercado bancário brasileiro	13
1.2	Objetivos	15
1.3	Justificativa	16
1.4	Organização do Trabalho	16
2	ASPECTOS CONCEITUAIS	17
2.1	Machine Learning	17
2.1.1	Redes Neurais	17
2.1.1.1	Generative Adversarial Network (GAN)	19
2.1.2	Algoritmos	20
2.1.2.1	Régressão Logística	20
2.1.2.2	Random Forest	21
2.1.2.3	Extreme Gradient Boost (XGBoost)	22
2.1.2.4	Multilayer Perceptron (MLP)	22
2.2	Visões Arquiteturais - RM-ODP	23
2.3	Business Process Model and Notations (BPMN)	24
2.4	Architecture Tradeoff Analysis Method (ATAM)	26
3	MÉTODO DO TRABALHO	28
4	ESPECIFICAÇÃO DO PROJETO	29
4.1	Visões Arquiteturais - RM-ODP	29
4.1.1	Visão de Negócio	29
4.1.1.1	Business Process Model and Notations (BPMN)	30
4.1.2	Visão de Requisitos Funcionais	31
4.1.3	Visão de Requisitos Não Funcionais	32
4.1.4	Visão de Mecanismos de Engenharia	32
4.1.5	Visão de Tecnologia	34
4.1.5.1	Geração de dados sintéticos	34
4.1.5.2	Algoritmos	35
5	DESENVOLVIMENTO DO TRABALHO	37
5.1	Dataset	37

5.1.1	Tratamento de dados	37
5.1.2	Dados sintéticos	38
5.1.3	Análise exploratória	38
5.2	Métricas	42
5.3	Algoritmos	44
5.3.1	Régressão Logística	44
5.3.2	Random Forest	46
5.3.3	Extreme Gradient Boosting (XGBoost)	47
5.3.4	Multilayer Perceptron (MLP)	50
5.4	Análises Comparativas	51
5.4.1	Architecture Tradeoff Analysis Method (ATAM)	52
5.4.1.1	Passo 1 – Coleta de Cenários	52
5.4.1.2	Passo 2 – Coleta de Requisitos, Restrições e Ambiente	52
5.4.1.3	Passo 3 – Descrição das Arquiteturas	53
5.4.1.4	Passo 4 – Análise dos Atributos Específicos	53
5.4.1.5	Passo 5 – Identificação de Sensibilidades	54
5.4.1.6	Passo 6 – Identificação de Trade-offs	54
5.4.1.7	Passo 7 – Priorização de Cenários	55
5.4.1.8	Passo 8 – Reanálise com Cenários Priorizados	55
5.4.1.9	Passo 9 – Apresentação dos Resultados e Recomendação	55
6	CONSIDERAÇÕES FINAIS	56
6.1	Conclusões	56
6.2	Contribuições	57
6.3	Perspectivas de Continuidade	57
	REFERÊNCIAS	59
	APÊNDICES	62
	APÊNDICE A – XAI - EXPLAINABLE AI	63
A.1	SHAP (SHapley Additive exPlanations)	64
A.2	Visualizações SHAP	64
A.2.1	Bar Plot	64
A.2.2	Summary Plot	64
A.2.3	Waterfall Plot	65
A.3	Resultados obtidos	65
A.3.1	Régressão Logística	66

A.3.2	Random Forest	69
A.3.3	XGBoost	72
A.3.4	Multilayer Perceptron (MLP)	74
A.4	Conclusão	76
APÊNDICE B – REPOSITÓRIO E REPRODUTIBILIDADE DOS RESULTADOS		77
ANEXOS		78
ANEXO A – DESCRIÇÃO DAS COLUNAS DO DATASET ORIGI- NAL		79

1 Introdução

1.1 Motivação

Os bancos, como instituições financeiras de grande relevância para a economia, estão expostos a uma ampla gama de riscos que podem comprometer sua estabilidade, imagem e sustentabilidade no mercado. A complexidade e a interconexão dos sistemas bancários tornam essas instituições vulneráveis a fatores internos e externos que podem impactar diretamente suas operações e sua relação com os clientes, investidores e reguladores. Com o avanço da tecnologia e as crescentes expectativas dos consumidores, a gestão de riscos tornou-se ainda mais crucial, exigindo que os bancos mantenham um monitoramento constante e implementem estratégias eficazes para mitigar potenciais ameaças. Entre os principais riscos existentes, estão ([BRASIL, 2025](#)):

- Risco de Crédito: Possibilidade de perdas decorrentes do não cumprimento de obrigações financeiras por contrapartes. O Banco Central estabelece limites por contraparte e para a carteira como um todo para controlar esse risco.
- Risco de Mercado: Potenciais perdas resultantes de flutuações em preços de mercado, como taxas de juros e câmbio. A gestão desse risco envolve a análise contínua das condições de mercado e a adoção de estratégias para mitigar impactos adversos.
- Risco de Liquidez: Risco de a instituição não conseguir honrar suas obrigações devido à falta de recursos líquidos. O gerenciamento desse risco inclui a manutenção de reservas adequadas e o monitoramento constante das necessidades de liquidez.
- Riscos Estratégicos: Perdas associadas a decisões estratégicas inadequadas ou à incapacidade de adaptar-se a mudanças no ambiente de negócios. A gestão desse risco requer planejamento estratégico robusto e flexibilidade para ajustar-se a novas circunstâncias.
- Riscos Operacionais: Perdas decorrentes de falhas em processos internos, pessoas, sistemas ou eventos externos. A mitigação envolve a implementação de controles internos eficazes e a promoção de uma cultura organizacional voltada para a gestão de riscos.

Entre todos os riscos enumerados, toma-se como escopo delimitar e focar nos **riscos de crédito** que os bancos estão sujeitos.

1.1.1 Dinâmicas do mercado

A dinâmica do risco de crédito nos bancos é fortemente influenciada por fatores econômicos e de mercado, como as variações na taxa de juros. Quando as taxas de juros estão em queda, os empréstimos e financiamentos se tornam mais acessíveis para os consumidores, impulsionando a demanda por crédito. Em contrapartida, períodos de alta nas taxas podem representar uma pressão adicional sobre os tomadores de crédito, especialmente os de perfil mais vulnerável, o que aumenta a possibilidade de inadimplência. Esse cenário cria um desafio adicional para os bancos, que precisam equilibrar o crescimento da carteira de crédito com o controle rigoroso dos riscos associados ao comportamento dos seus clientes.

Além disso, o aumento na demanda por crédito muitas vezes coincide com a entrada de novos competidores no setor financeiro. Instituições financeiras digitais, *fintechs* e outras *startups* estão desafiando o mercado tradicional ao oferecer condições atrativas e flexibilidade em suas operações de crédito. Esses novos competidores tendem a operar com margens mais apertadas, o que exerce pressão sobre os bancos para manterem sua competitividade. Esse ambiente de competição acirrada e margens reduzidas exige que os bancos revisem suas práticas de concessão de crédito e aprimorem suas metodologias de análise de risco para evitar perdas substanciais.

Diante desse cenário desafiador, os bancos têm adotado cada vez mais tecnologias avançadas e técnicas de modelagem para uma avaliação mais precisa do risco de crédito. Ferramentas de segmentação de clientes e análises preditivas baseadas em *behavior analytics* (análise do comportamento do cliente) tornaram-se fundamentais para diferenciar os perfis de risco e tomar decisões informadas. A segmentação de clientes permite aos bancos identificar padrões de comportamento, perfil de crédito e histórico de pagamento, possibilitando uma abordagem mais personalizada e seletiva na concessão de crédito. Dessa forma, conseguem calibrar suas ofertas, ajustar limites e condições de acordo com a resiliência financeira de cada grupo, reduzindo a exposição ao risco de inadimplência.

A modelagem de risco também é essencial para a construção de uma carteira de crédito mais robusta e resistente a variações econômicas. Com base em dados históricos e preditivos, os bancos podem criar modelos que antecipem o comportamento de diferentes segmentos de clientes diante de mudanças nas taxas de juros ou de condições macroeconômicas adversas. Essa abordagem permite que as instituições financeiras implementem estratégias de mitigação antes que o risco de crédito se materialize, aumentando a resiliência do banco e contribuindo para uma maior estabilidade financeira no longo prazo.

1.1.2 Perfil dos consumidores e do mercado bancário brasileiro

A população brasileira apresenta características econômicas, sociais e tecnológicas que influenciam diretamente o comportamento em relação ao mercado de crédito. Com um Produto Interno Bruto (PIB) de aproximadamente 1,9 trilhões de dólares em 2022, o Brasil ocupa a 11^a posição no *ranking* mundial, e o setor financeiro e bancário contribui com cerca de 7,5% desse valor ([IBGE, 2023](#)). Esse cenário coloca o país como uma economia relevante, onde o sistema financeiro desempenha um papel significativo, mas que ainda enfrenta desafios de inclusão e desenvolvimento econômico.

O Brasil possui uma população de cerca de 215 milhões de pessoas, das quais aproximadamente 43 milhões vivem em grandes centros urbanos com mais de um milhão de habitantes ([GOV, 2024](#)). A taxa de emprego está em torno de 57%, mas um aspecto notável é que 39% das vagas de trabalho provêm da economia informal, refletindo um cenário de vulnerabilidade econômica para uma parcela significativa da população ([IBGE, 2024b](#)). Além disso, levantamentos recentes mostram que aproximadamente 67% das famílias brasileiras possuem uma renda anual inferior a 35 mil reais, o que representa um desafio para o aumento do poder de compra e estabilidade financeira da população ([ALBUQUERQUE, 2022](#)).

Em termos de inclusão financeira, o Brasil apresenta uma das maiores taxas de bancarização da região, com 87% da população possuindo acesso a serviços bancários. Esse alto índice de bancarização é complementado pela média de 3,9 relacionamentos bancários por pessoa, um aumento em relação a 2018, quando a média era de 2,2 ([BRASIL, 2024](#)). Esse crescimento na relação com instituições financeiras demonstra uma evolução no acesso e uso dos serviços bancários, possibilitando maior diversificação e opções de crédito.

Outro aspecto importante é a conectividade digital, com cerca de 81% dos brasileiros utilizando internet móvel, o que facilita o acesso a serviços bancários digitais e *fintechs* ([IBGE, 2024a](#)). Esse dado revela que a população está cada vez mais integrada ao ambiente digital, o que cria um cenário favorável para o desenvolvimento de soluções tecnológicas no setor financeiro.

Essas perspectivas do perfil bancário da população brasileira ajudam a compreender o cenário dos bancos brasileiros frente a concentração de risco de crédito e composição de portfolio.

Quando analisando a Figura 1 (([ITAU, 2023](#)), ([SANTANDER, 2023](#)), ([BRADESCO, 2023](#)) e ([BRASIL, 2023](#))), percebe-se que bancos brasileiros enfrentam uma elevada concentração de risco de crédito em suas carteiras de ativos ponderados pelo risco. Em 2023, as quatro principais instituições financeiras (Itaú, Banco do Brasil, Bradesco e Santander) apresentaram uma parcela significativa de seus ativos exposta ao risco de crédito, com valores variando entre 77,9% e 84% dos ativos ponderados. Outras categorias

de risco, como risco operacional, risco de mercado e outros riscos, representam uma fração menor dos ativos, variando de 2% a 14%. Esses dados destacam a importância do risco de crédito como o principal componente de risco para os bancos no Brasil, o que exige uma gestão robusta e criteriosa para mitigar possíveis perdas decorrentes da inadimplência.

Tipos de risco com base em ativos ponderados pelo risco (2023, R\$ milhões)

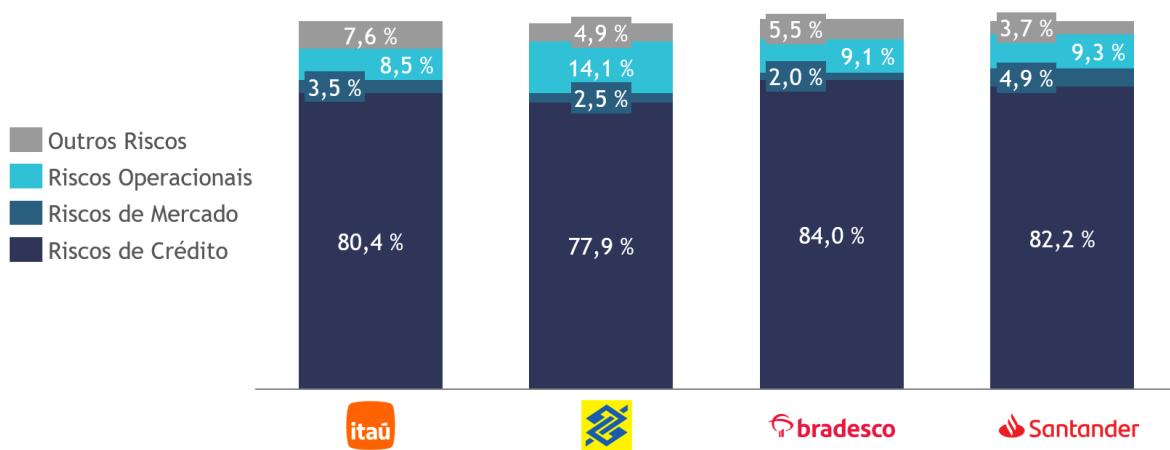


Figura 1 – Tipos de riscos em bancos

No contexto das receitas do setor bancário no Brasil em 2022, os empréstimos representam cerca de metade do "revenue pool" dos bancos de varejo, com destaque para os empréstimos não garantidos (como crédito pessoal e cheque especial) e empréstimos garantidos (como financiamento de veículos e hipotecas). Os cartões de crédito também desempenham um papel relevante, contribuindo com uma parcela expressiva da receita. Outros produtos, como depósitos, investimentos e taxas de serviço, complementam as fontes de receita dos bancos, embora em menor proporção. Essa estrutura de receita reflete a importância das linhas de crédito e cartões no portfolio de produtos dos bancos brasileiros, além da necessidade de diversificação para reduzir a dependência de produtos de crédito e mitigar riscos associados a eles.

Já no aspecto de inadimplência, a Figura 3 destaca que a inadimplência de crédito entre pessoas físicas no Brasil é consistentemente maior do que a inadimplência entre empresas (BRASIL, 2023). Desde o final de 2023, observa-se uma tendência de queda nas taxas de inadimplência em ambos os segmentos, mas a diferença entre eles persiste, com indivíduos apresentando uma taxa de inadimplência mais elevada. Esse comportamento pode ser atribuído à maior vulnerabilidade econômica das pessoas físicas, que frequentemente possuem menor resiliência financeira e estão mais expostas a instabilidades econômicas. Além disso, o crédito para consumidores tende a ter condições mais acessíveis, o que, apesar de facilitar o acesso, também aumenta o risco de inadimplência nesse grupo em comparação com as empresas, que geralmente possuem estruturas financeiras mais

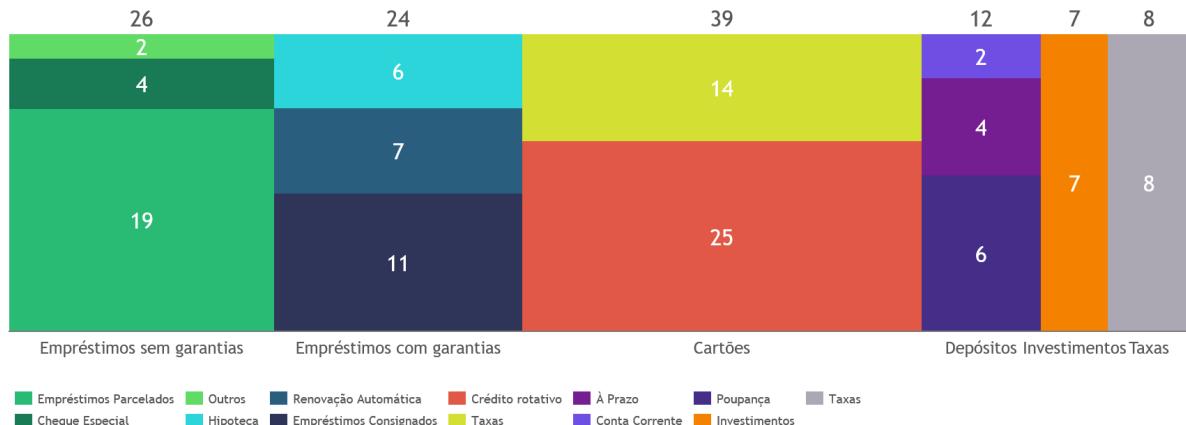


Figura 2 – Composição da receita em bancos

robustas e controle de risco mais rigoroso.

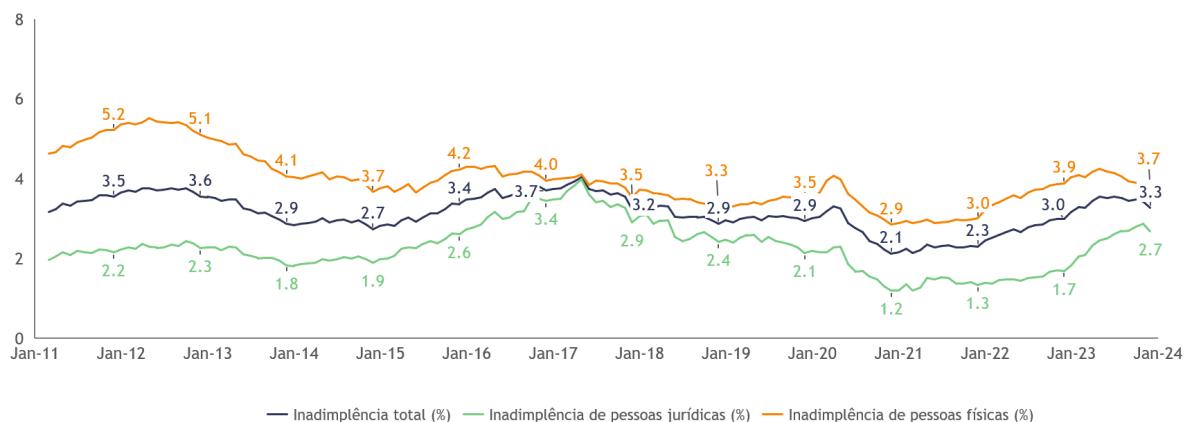


Figura 3 – Inadimplência

1.2 Objetivos

O objetivo deste trabalho é realizar uma análise comparativa entre os principais algoritmos de *machine learning* aplicados ao processo de concessão de crédito para pessoas físicas. Para isso, será utilizado um conjunto de dados que passará por etapas de tratamento e análise exploratória, a fim de garantir a consistência e qualidade das informações. Em seguida, os modelos selecionados serão treinados individualmente, com foco na geração de métricas de desempenho que refletem sua capacidade em tarefas de classificação de crédito. Por fim, será aplicado o método ATAM (Architecture Tradeoff Analysis Method), com o intuito de realizar uma análise comparativa abrangente entre os algoritmos, considerando não apenas o desempenho, mas também aspectos técnicos e operacionais relevantes para o contexto da concessão de crédito.

1.3 Justificativa

Conforme explorado em 1.1, a concessão de crédito é uma das principais fontes de receita para os bancos brasileiros, especialmente por meio de empréstimos, financiamentos e cartões de crédito. Contudo, esse segmento apresenta os maiores índices de inadimplência, o que aumenta o risco de crédito — o principal componente dos ativos ponderados pelo risco no país. Nesse cenário, tomar decisões de crédito mais precisas é crucial para reduzir perdas financeiras e fortalecer a estabilidade e competitividade do setor bancário. Este trabalho analisa e compara diferentes algoritmos de *machine learning* aplicados à concessão de crédito, com o objetivo de identificar os modelos mais eficazes na previsão de risco. Além disso, propõe uma análise comparativa utilizando o método ATAM (Architecture Tradeoff Analysis Method), que permite avaliar os modelos de forma técnica e qualitativa para sua aplicação no setor bancário.

Os trabalhos (SUHADOLNIK; UEYAMA; SILVA, 2023) e (CHANG et al., 2024) serviram como algumas das principais referências para a realização deste estudo, especialmente no que se refere ao uso de algoritmos de *machine learning* aplicados à concessão de crédito. No entanto, este trabalho se diferencia ao não se limitar ao desenvolvimento dos algoritmos, mas ao adotar uma abordagem mais ampla, com uma análise comparativa estruturada utilizando o método ATAM. Além disso, a especificação de projeto baseada em RM ODP (Reference Model of Open Distributed Processing) foi incorporada, oferecendo uma base robusta para o desenvolvimento dos algoritmos.

1.4 Organização do Trabalho

A organização deste trabalho é estruturada da seguinte forma:

- O Capítulo 2 apresenta os aspectos conceituais fundamentais sobre *machine learning*, RM ODP e ATAM, estabelecendo a base teórica necessária para o entendimento do trabalho.
- O Capítulo 3 descreve o método de trabalho adotado, explicando as etapas e fases.
- O Capítulo 4 detalha a especificação do projeto com base no modelo RM ODP, definindo a arquitetura e requisitos.
- O Capítulo 5 aborda o desenvolvimento dos algoritmos e a análise comparativa dos resultados obtidos.
- O Capítulo 6 traz as considerações finais, incluindo conclusões e lições aprendidas.

2 Aspectos Conceituais

2.1 Machine Learning

Aprendizado de máquina (*Machine Learning*) é uma subárea de estudo da inteligência artificial cujo objetivo é extrair conhecimentos de dados, de forma a desenvolver sistemas que aprendam com tais dados a partir de técnicas computacionais, acumulando experiência e permitindo o aperfeiçoamento contínuo do sistema de maneira autônoma ([GOOGLE, 2023a](#)). É uma área multidisciplinar, abordando matemática, estatística e probabilidade, e ciência da computação.

Por meio de técnicas como redes neurais e árvores de decisão, o aprendizado de máquina (ML) desenvolve modelos capazes de aprender a partir de exemplos, sem a necessidade de programação explícita. Para isso, se faz essencial grandes volumes de dados a serem consumidos por algoritmos de ML, permitindo que eles aprimorem seu desempenho por meio da experiência.

Os modelos de ML podem ser enquadrados em 3 categorias principais: aprendizado supervisionado, não supervisionado e por reforço:

- Aprendizado Supervisionado:

Modelo que utiliza dados estruturados para o treinamento, observando pares de entrada e saída previamente rotulados para inferir uma função de mapeamento de entrada-saída, minimizando o erro.

- Aprendizado Não Supervisionado:

Modelo usa dados não rotulados, buscando identificar padrões dentre os dados de entrada sem saber explicitamente as saídas esperadas. Desse modo, o modelo agrupa os dados com base em atributos identificados em padrões e distribuições aprendidas.

- Aprendizado por Reforço:

Modelo aprende a partir de uma interação com o ambiente, recendo recompensas ou punições para cada ação executada. O objetivo final do modelo é encontrar uma política que maximize as recompensas ao longo de sua execução.

2.1.1 Redes Neurais

Redes Neurais são modelos computacionais inspirados na estrutura e funcionamento do cérebro humano ([RUSSEL, 2013](#)). Esses modelos constituem um tipo de aprendizado

de máquina supervisionado, ou seja, as redes neurais usam dados de entrada-saída já conhecidos como fonte para o treinamento do modelo e seu aprendizado.

A estrutura básica de uma rede neural é composta por 3 camadas: (i) camada de entrada, (ii) camada oculta, responsável pelo processamento de informações, (iii) camada de saída, onde é gerada a saída do modelo.

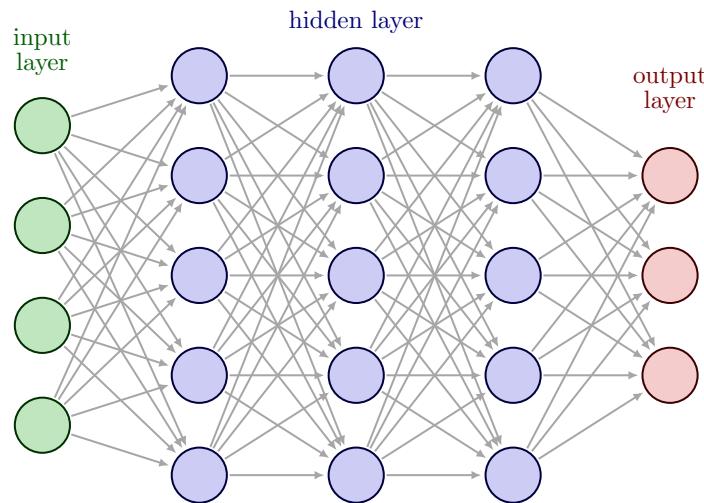


Figura 4 – Diagrama de uma Rede Neural

Cada camada é composta por neurônios, a unidade básica de uma rede neural, também chamados de *perceptrons*. O modelo matemático de um neurônio descreve o funcionamento desse com duas funções principais:

Função de Entrada Calcula a soma ponderada das entradas usando os pesos de cada ligação de entrada, além de um *bias* (viés). A fórmula da função é $z = \sum_{i=1}^n (x_i \cdot w_i) + b$, onde b é o viés, x_i são as entradas e w_i os pesos associados a cada entrada.

Função de ativação O resultado da função de entrada é passado para a função de ativação, a qual produz a saída do neurônio. Existem diversos tipos de funções de ativação, sendo as 4 principais: sigmoide, threshold, rectifier e tangente hiperbólica.

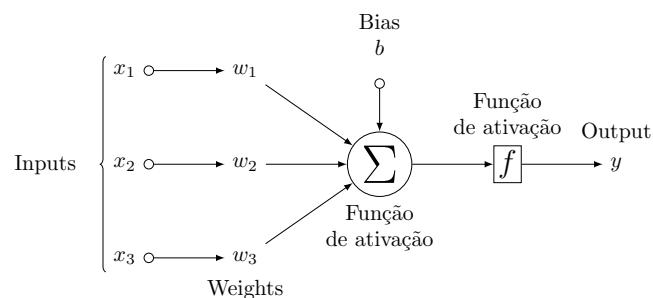


Figura 5 – Modelo matemático de um neurônio

Essas funções permitem que os neurônios realizem transformações não lineares sobre os dados, o que é fundamental para que a rede neural consiga aprender por experiência padrões complexos.

Existem diversos tipos de redes neurais, cada um com características estruturais específicas que os tornam mais adequados para determinados tipos de tarefas. Entre os principais, destacam-se as redes do tipo Perceptron Multicamadas (MLP), as Redes Neurais Convolucionais (CNN), as Redes Neurais Recorrentes (RNN), os *Autoencoders*, as Redes Adversárias Generativas (GANs) e os modelos baseados em *Transformers*.

2.1.1.1 Generative Adversarial Network (GAN)

As Redes Adversárias Generativas (GANs) são um tipo de modelo de aprendizado gerativo profundo composto por duas redes neurais que trabalham de forma adversária: o gerador e o discriminador (ACADEMY, 2022).

O gerador cria novos dados sintéticos, enquanto o discriminador avalia se esses dados são autênticos ou falsos, comparando-os com exemplos reais do conjunto de treinamento.

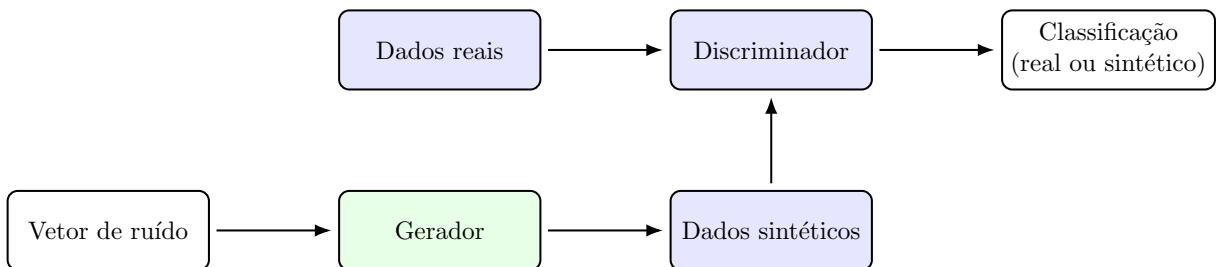


Figura 6 – Funcionamento de uma GAN

Esse processo (ilustrado na figura 6) estabelece uma interação competitiva, onde o gerador tenta enganar o discriminador ao produzir dados que se assemelham cada vez mais aos reais, enquanto o discriminador melhora sua capacidade de identificar falsificações.

O funcionamento de uma GAN ocorre por meio de um ciclo de *feedback*. O gerador parte de um vetor de ruído aleatório e utiliza técnicas como *upsampling* para criar dados, enquanto o discriminador processa tanto esses dados gerados quanto os reais, atribuindo probabilidades que indicam a autenticidade de cada entrada. Durante o treinamento, ambas as redes ajustam seus parâmetros para otimizar funções de perda opostas, em que os avanços de uma forçam a outra a melhorar. O resultado desse processo é um gerador que aprende a criar dados sintéticos altamente realistas, aproximando-se das características dos dados originais.

2.1.2 Algoritmos

2.1.2.1 Regressão Logística

A Regressão Logística é um algoritmo de aprendizado de máquina supervisionado, utilizado principalmente para problemas de classificação, apesar do termo "regressão" em seu nome. Diferentemente da regressão linear, que prevê valores contínuos, a regressão logística modela a probabilidade de um evento ocorrer, resultando em uma saída binária (por exemplo, sim/não, 0/1) ou categórica (AWS, 2025).

O funcionamento da Regressão Logística baseia-se na aplicação de uma função sigmoide (ou função logística) sobre uma combinação linear das variáveis de entrada. Essa função mapeia qualquer valor real para um intervalo entre 0 e 1, que pode ser interpretado como uma probabilidade. Em outras palavras, o modelo é linear em relação ao logit — isto é, ao logaritmo da razão entre as probabilidades de sucesso e insucesso — definido como:

$$\text{logit}(p) = \ln \left(\frac{p}{1 - p} \right)$$

A probabilidade é então obtida aplicando a função sigmoide sobre a combinação linear das variáveis:

$$p = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x_1 + \dots + \beta_n x_n)}}$$

Se a probabilidade calculada for superior a um determinado limiar (por exemplo, 0,5), a instância é classificada em uma classe; caso contrário, na outra. Os coeficientes do modelo são ajustados durante o treinamento, geralmente por meio do método de máxima verossimilhança, para minimizar o erro de classificação.

Entre suas vantagens, a Regressão Logística é relativamente fácil de implementar e interpretar, sendo computacionalmente eficiente, especialmente quando as classes são linearmente separáveis. Além disso, ela fornece uma probabilidade associada à classificação, o que pode ser útil para entender a confiança do modelo em suas previsões. Suas aplicações são vastas, incluindo, mas não se limitando a: previsão de inadimplência de crédito, diagnóstico médico (probabilidade de uma doença), marketing (probabilidade de um cliente clicar em um anúncio), e classificação de e-mails como spam ou não-spam. Contudo, suas limitações incluem a suposição de linearidade na relação entre as variáveis de entrada e o logit da variável de resposta, o que pode torná-la menos adequada para problemas com relações não lineares complexas entre os dados.

2.1.2.2 Random Forest

Random Forest é um modelo de aprendizado de máquina supervisionado baseado na criação de múltiplas árvores de decisão treinadas em subconjuntos aleatórios de dados, resultando em saídas que então são combinadas com base em critérios pré-definidos (SYSTEMS, 2007).

O algoritmo de *Random Forest* é um baseado em dois conceitos, sendo o primeiro o método *bagging*, no qual são criadas múltiplas amostras aleatórias do conjunto original de dados, podendo ocorrer a escolha de dados individuais em mais de uma amostra. Após a geração de várias amostras, os modelos são treinados de forma independente e os resultados são agregados para formar uma única previsão.

O segundo conceito são as árvores de decisão, que consistem em um método de aprendizado de máquina que utiliza uma estrutura hierárquica para modelar e representar decisões, buscando encontrar a melhor divisão para subconjuntos de dados. Uma árvore é composta por um nó raiz e nós folha. O nó raiz é o ponto inicial do processo de decisão e contém todo o conjunto inicial dos dados. A partir desse ponto, são feitas divisões sucessivas dos dados, guiadas por critérios que envolvem testes sobre as variáveis existentes no conjunto. Cada nó intermediário representa um desses testes ou condicionais, levando a resultados parciais que encaminham a decisão para outros níveis da árvore. Por fim, os nós folha são os terminais, indicando os resultados finais das decisões ou previsões realizadas pela árvore.

O *Random Forest* utiliza do *bagging* para criar árvores de baixa correlação entre si, escolhendo apenas um subconjunto aleatório das variáveis disponíveis em cada ponto de divisão. Ao final do treinamento, as árvores individuais produzem suas próprias previsões, que são então combinadas em uma previsão final consolidada. No caso de problemas de classificação, o *Random Forest* combina as previsões das árvores por meio de uma votação majoritária.

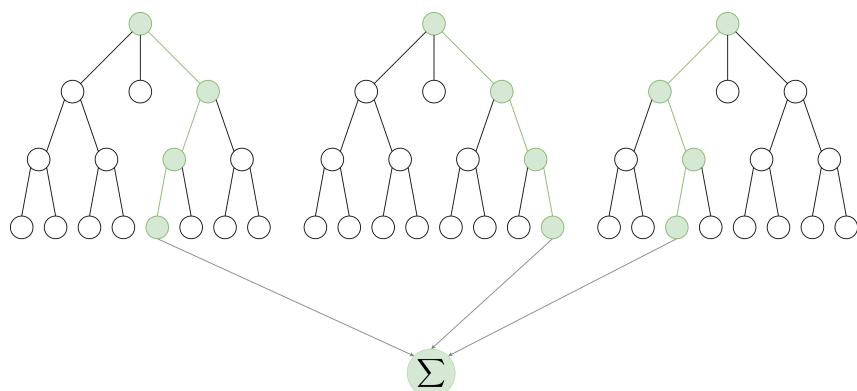


Figura 7 – Esquema do Random Forest

Entre as vantagens principais do *Random Forest* estão sua capacidade de reduzir

significativamente o overfitting, pois a agregação dos resultados de múltiplas árvores reduz a variância do modelo, além da possibilidade de avaliar a importância relativa das variáveis utilizadas no modelo. Contudo, o *Random Forest* geralmente apresenta maior complexidade computacional e menor interpretabilidade direta em comparação a uma única árvore de decisão.

2.1.2.3 Extreme Gradient Boost (XGBoost)

O XGBoost (eXtreme Gradient Boosting) é uma técnica de aprendizado supervisionado que utiliza árvores de decisão com reforço gradativo (*boosting*), onde cada árvore subsequente é construída para corrigir os erros das anteriores, minimizando os resíduos por meio da otimização de uma função de perda ([GOOGLE, 2023b](#)).

O *boosting* é um método de aprendizado que constrói modelos combinando várias árvores de decisão fracas sequencialmente, atribuindo pesos à saída de árvores individuais e, em seguida, peso maior às classificações incorretas da primeira árvore de decisão e uma entrada para a próxima árvore. Dessa forma, cada novo modelo corrige os erros cometidos pelos modelos anteriores.

Diferentemente do *bagging*, usado no *Random Forest*, que constrói árvores independentes, o *boosting* cria árvores de forma interdependente, aprimorando progressivamente a precisão da previsão ([IBM, 2024](#)).

Durante o treinamento, o modelo ajusta os pesos das observações e utiliza técnicas de regularização, que consistem em penalizar modelos muito complexos para evitar *overfitting* e melhorar a generalização dos dados. A predição final é feita combinando todas as árvores construídas, geralmente por meio de média casos de regressão ou votação para os de classificação.

A principal vantagem do XGBoost reside em sua eficiência computacional, uma vez que implementa paralelização no treinamento das árvores, reduzindo o tempo de execução. Além disso, a regularização ajuda a evitar *overfitting* e também permite ajustes mais finos de hiperparâmetros e suporte a diferentes tipos de dados. Apesar de sua alta precisão, o XGBoost possui algumas desvantagens, como a necessidade de ajuste fino dos hiperparâmetros para obter o melhor desempenho, maior custo computacional em comparação com modelos mais simples e maior dificuldade para interpretar e auditar resultados.

2.1.2.4 Multilayer Perceptron (MLP)

Multilayer Perceptron (MLP) é um modelo de aprendizado de máquina supervisionado baseado em redes neurais artificiais, composto por múltiplas camadas de neurônios interconectados, resultando em uma estrutura capaz de aprender padrões complexos a

partir dos dados ([LEARN, 2025](#)).

O MLP é baseado em uma arquitetura de camadas (como mostrado na Figura 4), onde os neurônios são organizados em três tipos principais: camada de entrada, que recebe os dados de entrada; camadas ocultas, que processam e extraem características; e camada de saída, que gera a previsão final do modelo.

Além disso, o treinamento ocorre por retropropagação do erro (*backpropagation*), um método de otimização que ajusta os pesos (letra w da Figura 5) das conexões entre os neurônios para minimizar a diferença entre a saída prevista e a real. Esse processo ocorre através do cálculo do erro da previsão e da propagação desse erro de volta pela rede, permitindo ajustes nos pesos por meio de algoritmos como o gradiente descendente.

Cada neurônio do MLP aplica uma função de ativação não linear, como ReLU ($f(x) = \max(0, x)$) ou sigmoide, possibilitando que a rede aprenda relações não triviais nos dados, como mostrado na seção [2.1.1](#). O uso de múltiplas camadas permite a modelagem de padrões altamente não lineares, tornando o MLP eficaz para diversas tarefas, como classificação e regressão.

Entre as principais vantagens do MLP estão sua capacidade de capturar padrões complexos e sua flexibilidade para diferentes tipos de dados. No entanto, o modelo pode ser computacionalmente custoso e requer um ajuste adequado dos hiperparâmetros para evitar problemas como *overfitting* ou treinamento ineficiente. Outra desvantagem é seu funcionamento de "caixa opaca", fator que diminui a interpretabilidade do modelo.

2.2 Visões Arquiteturais - RM-ODP

O Modelo de Referência para Processamento Distribuído Aberto (RM-ODP – *Reference Model of Open Distributed Processing*) é um padrão de modelagem de arquiteturas em processos distribuídos, desenvolvido com o intuito de fornecer uma base comum para o projeto e desenvolvimento de sistemas distribuídos abertos ([PUTMAN, 2000](#)).

Definido pela norma ISO/IEC 10746, o modelo busca estabelecer um conjunto coerente e abrangente de conceitos essenciais que fornece uma estrutura bem desenvolvida para a estruturação de especificações para sistemas distribuídos ([ISO/IEC, 2009](#)).

O RM-ODP se caracteriza pela abordagem baseada em visões, proporcionando diferentes níveis de abstração e preocupação para organização e desenvolvimento dos sistemas. Essas visões são:

- Visão da Empresa:

Define o escopo, políticas, objetivos e requisitos do sistema no contexto do negócio ou ambiente operacional, fornecendo uma perspectiva das necessidades do usuário

e do propósito geral do sistema. Inclui também aspectos organizacionais como papéis, responsabilidades, fluxos de trabalho e interações com stakeholders externos, ajudando a alinhar o sistema com os objetivos estratégicos do negócio.

- Visão da Informação :

Descreve a estrutura e o fluxo das informações gerenciadas pelo sistema, abordando como a informação é armazenada, processada e compartilhada entre os diferentes componentes e subsistemas. Essa perspectiva enfatiza os modelos de dados, esquemas de armazenamento, técnicas de representação da informação e semântica dos dados, essenciais para garantir coerência e integridade das informações distribuídas.

- Visão Computacional:

Especifica a funcionalidade do sistema, considerando as interações entre objetos distribuídos, os serviços oferecidos e as interfaces necessárias para que os diversos componentes interajam e cooperem. Essa visão foca principalmente nas operações e comportamentos esperados dos objetos, identificando claramente os pontos de comunicação, contratos de serviço e protocolos utilizados para a colaboração eficaz entre componentes distribuídos.

- Visão de Engenharia:

Trata dos mecanismos e recursos tecnológicos necessários para a distribuição das funções e objetos identificados na visão computacional. Aborda a infraestrutura, protocolos, tecnologias de comunicação e mecanismos de gerência, controle e monitoramento necessários à implementação efetiva e eficiente do sistema distribuído. A visão de engenharia preocupa-se também com aspectos de desempenho, escalabilidade, confiabilidade, tolerância a falhas, segurança e interoperabilidade.

- Visão de Tecnologia:

Define a implementação prática do sistema, incluindo escolhas específicas de tecnologias, linguagens de programação, plataformas, *hardware* e protocolos concretos que tornarão possível o funcionamento real do sistema. Esta visão trata também de decisões técnicas detalhadas, como questões relativas à instalação, configuração, manutenção e otimização dos recursos tecnológicos escolhidos.

2.3 Business Process Model and Notations (BPMN)

A notação BPMN (*Business Process Model and Notation*) é uma linguagem gráfica padronizada internacionalmente utilizada pra modelar processos de negócio, permitindo uma visualização clara e comprehensível das atividades e fluxos envolvidos. Desenvolvida inicialmente pela *Business Process Management Initiative* (BPMI) em 2004 e atualmente

mantida pelo Object Management Group (OMG), a BPMN tem como principal objetivo facilitar a comunicação entre os diversos *stakeholders*, incluindo analistas de negócios, desenvolvedores e gestores ([CHINOSI, 2012](#)).

O BPMN é baseada em fluxogramas e é composta por 4 categorias principais de elementos ([OMG, 2011](#)):

1. Objetos de Fluxo:

Principais elementos gráficos para descrever o comportamento de um processo de negócio.

- Eventos: representam ocorrências que afetam o fluxo do processo, podem ser de início, intermediários ou de fim.
- Atividades: tarefas ou subprocessos a serem executados no processo.
- *Gateways*: representam pontos de decisão ou ramificações no fluxo de processo.

2. Objetos de Conexão:

Elementos que conectam os objetos de fluxo entre si e com outros tipos de informação.

- Fluxos de sequência: define a ordem de execução das atividades dentro do processo.
- Fluxos de mensagem: indica a comunicação entre diferentes participantes ou entidades no processo.
- Associações: relaciona artefatos ou informações adicionais às atividades ou eventos.

3. *Swimlanes* (raias):

Constituem a representação gráfica de participantes.

4. Artefatos

Fornecem informações adicionais sobre os processos.

- Objeto de dados: fornece informações sobre os dados utilizados ou produzidos durante o processo.
- Grupo: agrupa elementos relacionados dentro do diagrama para fins de organização ou documentação.
- Texto Adicional: adiciona comentários ou explicações adicionais ao diagrama.

2.4 Architecture Tradeoff Analysis Method (ATAM)

O *Architecture Trade-off Analysis Method* (ATAM) é uma técnica de análise sistemática usada para a tomada de decisões arquiteturais em sistemas de software. Seu objetivo principal é avaliar o quanto uma arquitetura satisfaz objetivos de qualidade e identificar *trade-offs* existentes entre os diferentes atributos de qualidade desejados em uma arquitetura, tais como desempenho, segurança, confiabilidade, escalabilidade e manutenibilidade (KAZMAN MARK KLEIN, 2000). Em suma, o ATAM avalia as consequências das decisões arquiteturais à luz dos requisitos dos atributos de qualidade, de forma a detectar e evitar riscos intrínsecos à arquitetura, como realizado em (HAYASHI; ARAKAKI; RUGGIERO, 2020).

O processo do ATAM consiste em nove passos, agrupados em quatro etapas:

- **Apresentação**

1. Apresentar o ATAM: O método é descrito aos participantes e *stakeholders*.
2. Apresentar os *Business Drivers*: O gerente de projeto descreve os objetivos de negócio impulsionadores do desenvolvimento, e portanto, quais os requisitos a arquitetura deve cumprir.
3. Apresentar Arquitetura: A arquitetura proposta é descrita, focando em explicar como ela atende aos *business drivers*.

- **Investigação e Análise**

4. Identificar abordagens arquiteturais: As abordagens arquiteturais são identificadas pelo arquiteto, mas não são analisadas. São levantados os principais tipos de arquiteturas de referência que podem atender os requisitos de qualidade.
5. Gerar árvore utilitária de atributo de qualidade: São elicitados os fatores de qualidade necessários ao sistema como desempenho, disponibilidade, segurança, modificabilidade, indicados em nível de cenários anotados com estímulos e respostas, e priorizados.
6. Analisar abordagens arquiteturais: Com base nos fatores priorizados no passo anterior, as abordagens arquiteturais são indicadas e analisadas de modo a endereçar esses fatores. Durante este passo são identificados riscos arquiteturais, pontos de sensibilidade e pontos de *trade-off*.

- **Testes**

7. *Brainstorm* e priorização de cenários: A partir dos cenários descritos no passo da árvore de utilidade, esses são aprofundados com base nas interações das

partes interessadas com o sistema. Também ocorre a priorização e eliminação de cenários.

8. Analisar abordagens arquiteturais: Este passo repete o passo 6, mas agora com os cenários finais escolhidos e priorizados.

- **Resultados**

9. Apresentar resultados: Com base nas informações coletadas durante o processo, a equipe apresenta as descobertas aos *stakeholders* presentes e, gera um relatório detalhando essas informações juntamente com estratégias propostas.

A aplicação do ATAM permite analisar, revisar e priorizar requisitos do sistema, além de mapear elementos necessários para a arquitetura proposta. O método permite também uma avaliação de benefícios e desvantagens das decisões arquiteturais feitas durante o processo do ATAM ([LIMA, 2016](#)).

3 Método do trabalho

O método de trabalho adotado neste projeto é dividido em várias fases, cada uma com objetivos específicos. As fases são:

- Especificação do projeto: Utilizando o modelo RM ODP, esta fase define a arquitetura do projeto a partir das cinco visões arquiteturais, estabelecendo os requisitos e as bases para o desenvolvimento.
- Geração de dados sintéticos: Com o objetivo de expandir o *dataset*, esta fase foca em aplicar técnicas de redes neurais para gerar dados sintéticos.
- Definição de métricas de desempenho: A definição das métricas de desempenho é feita com base nas características do *dataset* e em uma consulta a referências científicas, estabelecendo quais indicadores serão acompanhados para avaliar o desempenho dos algoritmos.
- Desenvolvimento de algoritmos: Nesta fase, são treinados os modelos selecionados, realizados testes e colhidas as métricas de desempenho previamente definidas.
- Comparação de desempenho: Utilizando o método ATAM (*Architecture Tradeoff Analysis Method*), esta fase compara os algoritmos em termos de desempenho e analisa os *trade-offs* entre eles, considerando tanto aspectos quantitativos quanto qualitativos dos resultados obtidos.

4 Especificação do Projeto

4.1 Visões Arquiteturais - RM-ODP

Utilizando e aplicando os conceitos de Visões Arquiteturais descritos em [2.2](#) para o problema proposto em específico, é possível desenhar as especificações do projeto a ser desenvolvido.

As visões serão segmentadas em cinco: Visão de Negócio, Visão de Requisitos Funcionais, Visão de Requisitos Não Funcionais, Visão de Mecanismos de Engenharia e Visão de Tecnologia. Tal segmentação é baseada na flexibilidade oferecida pela ISO/IEC 10746 em conjunto com a IEEE 1471 e ISO/IEC 42010.

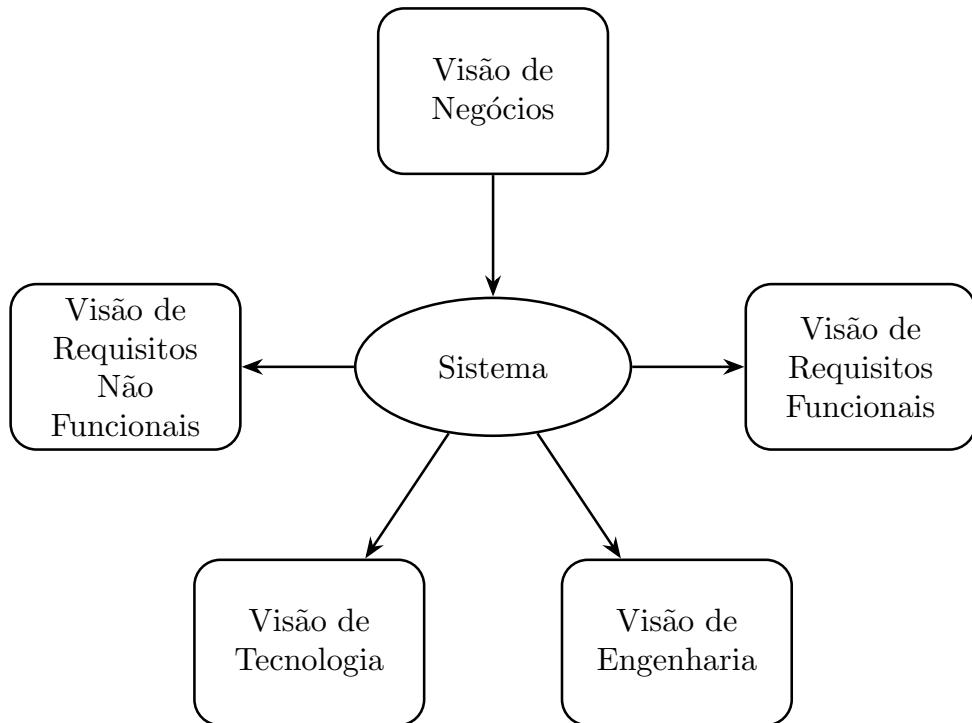


Figura 8 – Visões Arquiteturais

4.1.1 Visão de Negócio

A visão de negócio do sistema de avaliação de crédito busca alinhar suas funcionalidades com os principais objetivos organizacionais, especialmente no que diz respeito à precisão e segurança na concessão de crédito. O sistema é projetado para tornar a distribuição de crédito mais assertiva, com foco em reduzir perdas financeiras causadas por inadimplência. Esse alinhamento é essencial para garantir que a instituição minimize riscos

e otimize sua carteira de crédito, proporcionando uma base sólida para uma operação financeira mais sustentável, menos exposta a perdas e ainda assim rentável.

Além disso, o sistema deve atender a políticas e regras de negócio rigorosas, que exigem uma análise de crédito precisa e rápida. Essa agilidade é fundamental para se adaptar ao ritmo competitivo do mercado, permitindo que os clientes recebam respostas com mais rapidez e assertividade. Como parte dessas diretrizes, o banco precisa manter altos padrões de segurança e cuidado na gestão de dados, considerando que lida com informações sensíveis de muitos clientes. Assim, a implementação de robustas práticas de governança de dados e conformidade é indispensável, garantindo que a privacidade e a segurança dos dados sejam mantidas em todas as etapas do processo.

A aplicação de algoritmos para análise de crédito contribui diretamente para alcançar resultados estratégicos e operacionais. Um sistema mais assertivo e eficiente não apenas reduz a carga de trabalho do setor de inadimplência, mas também torna o processo de análise de crédito mais justo e ágil. Isso traz benefícios significativos para a experiência do cliente, que, ao acessar o serviço, encontra um processo claro e simplificado, aumentando a satisfação e fidelização ao serviço. Ao mesmo tempo, a organização fortalece sua imagem ao oferecer um serviço mais confiável e moderno, ajustado às demandas dos usuários e do mercado.

Um ponto importante a ser considerado são as nuances de assertividade na concessão de crédito para dois grupos de clientes: 1) os bons pagadores que não receberam crédito e 2) os maus pagadores que receberam crédito. Ambos os cenários representam desafios para o banco, mas o impacto de conceder crédito a um mau pagador é significativamente maior do que o de negar crédito a um bom pagador. Portanto, do ponto de vista do negócio, é mais crucial monitorar o segundo caso do que o primeiro.

4.1.1.1 Business Process Model and Notations (BPMN)

Para definir e estabelecer as etapas do fluxo de negócio e entender oportunidades de gerar valor, utiliza-se como metodologia o *Business Process Model and Notation - BPMN*, especificado na Figura 9.

O BPMN (definido com mais detalhes em 2.3) é uma notação gráfica padrão para modelar processos de negócios. Facilita a comunicação entre áreas técnicas e de negócio, promovendo análise, melhoria e automação de fluxos de trabalho.

O fluxo de negócio contém três principais partes: o cliente, o intermediário (por exemplo, um funcionário) e o sistema avaliador do crédito em si. O processo em si passa por algumas fases:

- Pedido de análise e envio de informações pelo cliente

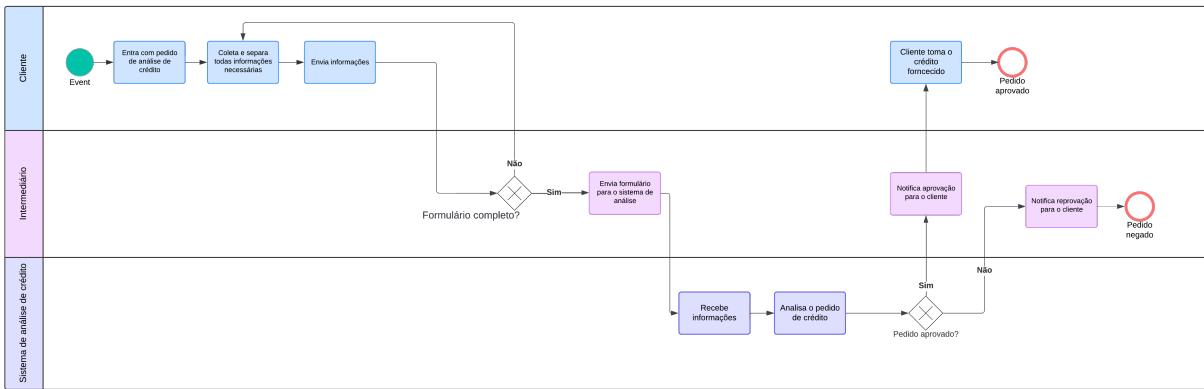


Figura 9 – BPMN

O cliente, ao desejar fazer um pedido de crédito ao banco, deve ser capaz de iniciar uma solicitação de análise de crédito. Para isso, ele deve fornecer dados essenciais para a tomada de decisão do banco. Essas informações podem incluir idade, sexo, estado civil, situação empregatícia, e se possui casa própria, entre outros dados relevantes. É importante que todos os dados necessários para a análise sejam fornecidos.

- Análise do risco de crédito

Com as informações do cliente em mãos, o sistema de análise de crédito avalia o perfil financeiro do solicitante e estima a probabilidade de ele ser um bom pagador. O sistema considera fatores internos, como a exposição ao risco desejada pela instituição, para decidir se o crédito será aprovado ou negado. Esse processo utiliza algoritmos que cruzam dados e aplicam modelos preditivos, aumentando a precisão na tomada de decisão. É nessa etapa que está o foco do projeto, que visa se aprofundar de maneira estruturada no desempenho de uma seleção de algoritmos.

- Comunicação do resultado e finalização

Após a análise, o portal comunica o resultado ao cliente. Em caso de aprovação, o cliente pode acessar o valor do empréstimo solicitado, e o processo é concluído. Caso o pedido seja negado, o banco também deve informar o cliente, encerrando o processo de forma clara e transparente. Essa última etapa garante que o cliente tenha uma experiência completa e compreenda o resultado de sua solicitação.

4.1.2 Visão de Requisitos Funcionais

Com base nas necessidades do negócio e focando no escopo do projeto (análise comparativa de algoritmos de concessão de crédito), pode-se identificar os seguintes requisitos funcionais:

- **RF01 - O algoritmo deve ser alimentado com dados** específicos do cliente e que sejam necessários para a análise de crédito (ex. idade, estado civil, renda).
- **RF02 - O algoritmo deve buscar como resultado analisar se o cliente é um bom pagador ou mau pagador**, de acordo com o resultado do algoritmo executado.
- **RF03 - O algoritmo deve retornar métricas de avaliação**, como acurácia, precisão e sensibilidade, além da matriz de confusão correspondente.

4.1.3 Visão de Requisitos Não Funcionais

Com base nas necessidades do negócio e focando no escopo do projeto (análise comparativa de algoritmos de concessão de crédito), pode-se identificar os seguintes requisitos não funcionais:

- **RNF01 - Acurácia do algoritmo** deve ser a maior possível, estando entre 80% e 99% ([CHANG et al., 2024](#)).
- **RNF02 - Tempo de treinamento** do modelo idealmente deve ser baixo. Embora o treinamento não aconteça todos os dias, é um requisito a ser considerado.
- **RNF03 - Disponibilidade do sistema e do algoritmo** deve ser garantida durante o horário comercial e preferencialmente 24h.
- **RNF04 - Segurança do sistema** deve garantir que os dados sejam criptografados e só pessoas autorizadas tenham acesso.
- **RNF05 - Conformidade legal do sistema** deve estar de acordo com a LGPD existente.

Embora os últimos três requisitos não funcionais enumerados acima sejam de extrema importância, eles não serão escopo ou aprofundados neste trabalho. O projeto aqui desenvolvido tem como objetivo desenvolver algoritmos, coletar métricas de desempenho e estruturar uma abordagem comparativa.

4.1.4 Visão de Mecanismos de Engenharia

Para o desenvolvimento dos algoritmos de análise de crédito a serem comparados posteriormente, os seguintes componentes de engenharia devem ser estruturados:

- **Geração de dados sintéticos**

A geração de dados sintéticos consiste em criar conjuntos de dados artificialmente para simular cenários reais ou suprir a falta de dados disponíveis. Essa etapa é crucial em situações onde dados reais são escassos, sensíveis ou difíceis de obter.

- **Treinamento do algoritmo**

O treinamento do algoritmo é o processo de ajustar os pesos e parâmetros a fim de minimizar a diferença entre as previsões do modelo e os valores reais. Durante essa etapa, o modelo aprende padrões presentes nos dados de treinamento, com o objetivo de generalizar para novos dados. Configurações como escolha da arquitetura da rede, funções de ativação e hiperparâmetros desempenham um papel essencial.

- **Testes**

Após o treinamento, o modelo deve ser avaliado em um conjunto de testes para verificar sua capacidade de generalização. Nessa etapa, os dados de teste são processados pelo modelo para analisar seu comportamento em novas entradas. Esse processo é fundamental para identificar falhas e garantir que o modelo funcione conforme o esperado antes de sua validação.

- **Validação dos resultados**

A validação envolve uma análise mais profunda do desempenho do modelo, verificando sua robustez e possíveis falhas. Problemas como *overfitting*, quando o modelo se ajusta excessivamente aos dados de treino, e *underfitting*, quando ele não aprende padrões relevantes, devem ser identificados e corrigidos. Essa etapa garante que o modelo seja confiável para aplicação prática.

- **Consolidação dos resultados**

Nesta etapa, as métricas de desempenho dos diferentes modelos avaliados são reunidas e organizadas de forma estruturada. Esse processo facilita a análise posterior, permitindo uma visão clara do comportamento de cada algoritmo nos testes realizados. A consolidação garante que todas as informações relevantes estejam disponíveis para suportar análises comparativas.

Para a análise comparativa, os seguintes algoritmos foram selecionados, estudados, implementados e consolidados, tomando-se o cuidado de escolher algoritmos de naturezas diversas:

1. Regressão Logística
2. *Random Forest*
3. *Extreme Gradient Boosting* (XGBoost)
4. *Multilayer Perceptron* (MLP)

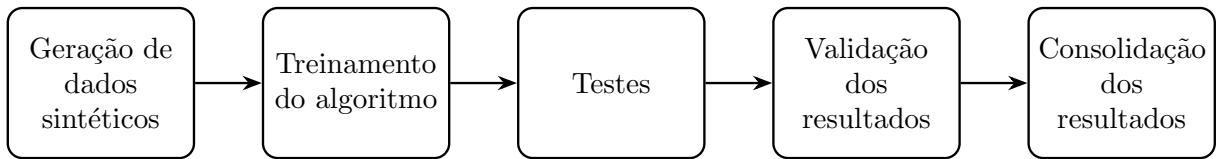


Figura 10 – Fluxo de Processos de Engenharia

4.1.5 Visão de Tecnologia

Pensando na implementação do projeto, essa visão foca em escolher a tecnologia a ser utilizada em cada componente dependendo da necessidade mapeada.

4.1.5.1 Geração de dados sintéticos

Para testar diferentes algoritmos de análise de crédito bancário, uma primeira necessidade é ter um conjunto de dados para trabalhar. Entretanto, este é um contexto onde dados reais são limitados, sensíveis ou sujeitos a restrições legais. Portanto, uma alternativa a ser usada é a geração de dados sintéticos a partir de um conjunto de dados menor. O algoritmo escolhido para esse tarefa foi o *Generative Adversarial Network*, explicado conceitualmente em 2.1.1.1.

A YData ([YDATA, 2020](#)) é uma empresa que oferece um conjunto de ferramentas voltadas para a geração, aprimoramento e análise de dados sintéticos, com foco especial em dados tabulares. Seu principal produto, YData Synthetic, é uma biblioteca em *Python* que facilita a criação de dados sintéticos realistas, preservando as propriedades estatísticas e estruturais dos dados originais. Ela suporta diversos tipos de algoritmos generativos de IA, como GAN, CGAN, CTGAN, WGAN, entre outros.

A YData Synthetic se destaca pela facilidade de uso, permitindo a geração de dados sintéticos de alta qualidade sem necessidade de configurações complexas. Inclusive, a empresa oferece a YData Fabric ([YDATA..., 2020](#)), uma interface UI que contém a solução de geração de dados sintéticos.

Além disso, sua capacidade de preservar padrões e dependências dos dados originais torna os dados sintéticos mais representativos e úteis para modelos de *machine learning*, especialmente em cenários de desbalanceamento de classes ou escassez de dados reais. No entanto, a biblioteca pode possuir alto custo computacional, especialmente ao treinar modelos para grandes volumes de dados, e sua eficácia depende da qualidade e diversidade dos dados originais.

Apesar das limitações da biblioteca, entende-se que ela atende a todas as necessidades do *dataset* a ser expandido, principalmente no aspecto de suporte a dados mistos e capacidade de lidar com desbalanceamento dos dados.



Figura 11 – YData

4.1.5.2 Algoritmos

Os algoritmos escolhidos para serem implementados e analisados comparativamente são Regressão Logística, *Random Forest*, XGBoost e *Multilayer Perceptron* (MLP). Os algoritmos escolhidos variam em termos de tipos de modelo, complexidade, tempo de treinamento, tendências de *overfitting* e facilidade de interpretação, e por isso compõem um conjunto de ferramentas interessante de serem avaliadas comparativamente. Para o desenvolvimento dos algoritmos citados acima, optou-se por utilizar principalmente a biblioteca Scikit-learn ([PEDREGOSA et al., 2011](#)), com exceção do XGBoost, que foi implementado utilizando a biblioteca XGBoost.

A biblioteca Scikit-learn é amplamente reconhecida como uma das ferramentas mais robustas e acessíveis para o desenvolvimento de aplicações de aprendizado de máquina em *Python*. Ela oferece uma vasta gama de algoritmos e ferramentas para tarefas de classificação, regressão, agrupamento, redução de dimensionalidade e validação de modelos. Baseada em bibliotecas de alto desempenho, como NumPy, SciPy e matplotlib, a Scikit-learn apresenta uma interface simples e coerente, que facilita sua integração em projetos de pesquisa e aplicações industriais. Além disso, sua documentação abrangente e sua comunidade ativa tornam a biblioteca uma escolha ideal.

A Scikit-learn também se destaca pela compatibilidade com outros pacotes do ecossistema *Python* e pela facilidade de uso em pipelines de aprendizado de máquina. A biblioteca oferece implementações eficientes de algoritmos amplamente utilizados, como máquinas de vetor de suporte (SVMs), random forest, regressão logística, k-vizinhos mais próximos (k-NN), entre outros. Apesar de não ser focada em redes neurais profundas, que são melhor abordadas por bibliotecas como TensorFlow e PyTorch, a Scikit-learn é ideal para construir, treinar e avaliar modelos de aprendizado supervisionado e não supervisionado.

A biblioteca XGBoost, por sua vez, foi projetada especificamente para oferecer uma implementação otimizada do algoritmo Extreme Gradient Boosting, com foco em desempenho e escalabilidade. Ela fornece suporte nativo para paralelismo, regularização e ajuste fino de hiperparâmetros, sendo amplamente adotada em competições e aplicações práticas de aprendizado de máquina devido à sua eficiência e alta capacidade preditiva.



Figura 12 – Scikit Learn

5 Desenvolvimento do Trabalho

5.1 Dataset

Uma parte bastante relevante do projeto desenvolvido é a manipulação do *dataset*. Para isso, foi utilizado um dataset de dados bancários desenvolvido por Hans Hofmann e disponível no Statlog ([HOFMANN, 1994](#)).

O *dataset* original contém 1000 entradas com 20 colunas de atributos categóricos ou numéricos. Cada linha representa uma pessoa que toma crédito de um banco. Cada pessoa é classificada como "bom pagador" ou "mau pagador", dependendo do conjunto de características levantadas. As informações elencadas no *dataset* estão disponíveis na tabela 1. No Anexo A há uma relação mais detalhada do *dataset* original com todas as colunas disponíveis e os valores que cada uma pode ter atribuído.

Coluna	Tipo	Descrição
sex	Categórica	Sexo
marital_status	Categórica	Estado civil
age	Numérica	Idade
n_of_liables	Numérica	Número de dependentes
job	Ordinal	Trabalho
foreign_worker	Numérica	Trabalhador estrangeiro
present_employee_since	Ordinal	Tempo no trabalho atual
telephone	Numérica	Possui telefone
housing	Categórica	Moradia atual
present_residence_since	Numérica	Tempo na residência atual
property	Categórica	Tipo de propriedade possuída
checking_account	Ordinal	Conta corrente
savings	Ordinal	Poupança
purpose	Categórica	Finalidade
credit_history	Numérica	Nota de histórico de crédito
duration	Numérica	Duração em meses do empréstimo/crédito
credit_amount	Numérica	Quantidade de crédito em DM (marco alemão)
guarantors	Categórica	Quantidade de fiadores
other_installment_plans	Categórica	Outros financiamento parcelados
credits_at_bank	Numérica	Número de créditos no banco
risk	Numérica	Risco

Tabela 1 – Colunas do dataset

5.1.1 Tratamento de dados

Com o objetivo de preparar o *dataset* para ser manipulado nos algoritmos escolhidos, é necessário tratar os dados presentes. As colunas foram separadas em três categorias,

conforme enumerado na tabela 1:

- Numéricas: representam medidas ou contagens, podendo ser discretos ou contínuos;
- Categóricas: valores qualitativos que representam categorias sem ordem específica;
- Ordinais: tipo de dado categórico com uma ordem definida. Para trabalhar com esse tipo de dado

Dados do tipo categórico precisam ser transformados em variáveis *dummies* (ou indicadoras/binárias). Elas são úteis em regressões lineares para representar variáveis categóricas. Cada categoria se torna uma nova coluna com valores 0 ou 1, indicando sua presença ou ausência em cada observação.

5.1.2 Dados sintéticos

Com o objetivo de testar também a performance dos algoritmos com grandes volumes de dados (para esse trabalho, 1 milhão de linhas), foram aplicadas ao *dataset* original técnicas de *Generative Adversarial Network* (GAN) utilizando a biblioteca descrita em 4.1.5.1. O resultado final foi um *dataset* de 1 milhão de linhas que replica o perfil de dados encontrado no *dataset* original.

5.1.3 Análise exploratória

Para entender melhor a composição do *dataset* utilizado e poder auxiliar na verificação dos resultados obtidos, foram performadas algumas análises exploratórias.

Conforme indicado na figura 13, o *dataset* trabalhado possui um comportamento de distribuição de risco parecido entre os diferentes grupos de idade. Isso indica que nenhum dos grupos de idade se destaca por ter um risco muito maior quando comparado a outros. Todos os grupos possuem uma taxa de mau pagadores de 20% a 30% do total da amostra. Além disso, também pode-se observar que idades entre 18 e 45 anos tem maior participação no dataset.

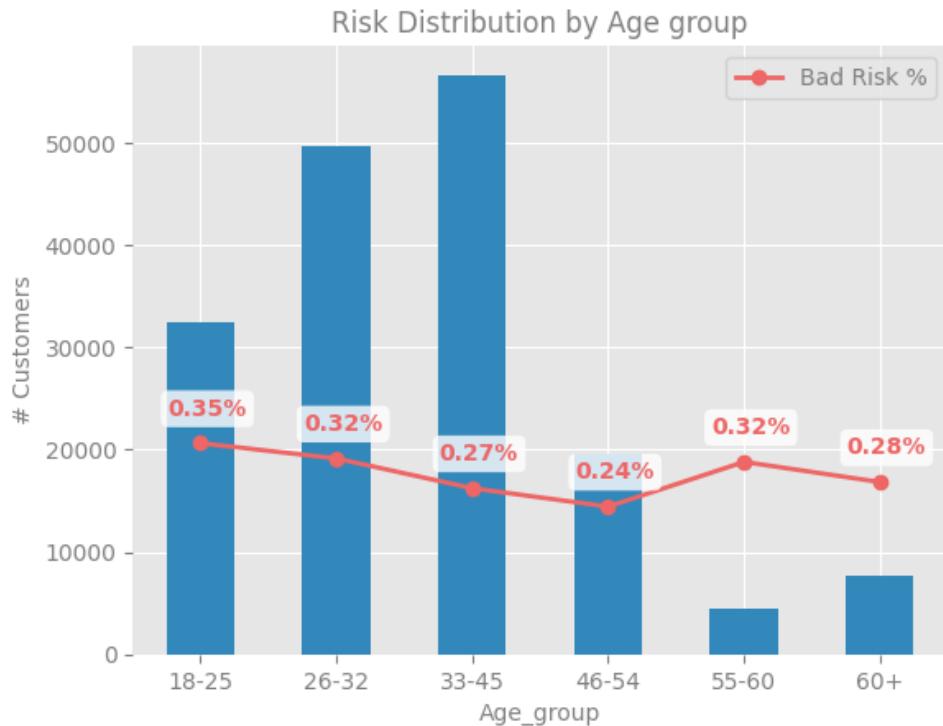


Figura 13 – Distribuição de risco por grupo de idade

Em termos de distribuição de sexo, percebe-se que o dataset é majoritariamente masculino, embora esse grupo tenha uma taxa de mau pagadores de 0,27% vs 0,35% para mulheres (Figura 14).



Figura 14 – Distribuição de risco por sexo

Outro *insight* interessante observado no *dataset* pode ser verificado na figura 15. A linha vermelha representa que porcentagem daquele grupo de crédito (isto é, pessoas que historicamente pediram por um valor dentro daquela faixa) na base é um mau pagador. A linha azul mostra uma visão acumulativa da quantidade de mau pagadores presentes na base segmentados também por grupo de crédito. Por fim, a linha verde pontilhada indica apenas o marco acumulativo da linha azul referente a 80% da base.

Isso significa que, a maior parte das pessoas consideradas mau pagadores da base usualmente pedem valores de crédito de até 6750 DM. Porém, o comportamento da linha vermelha quando chega-se em altos valores de crédito indica que, se uma pessoa pedir altas quantias de crédito, a chance de ela ser uma mau pagadora é maior (aproximadamente de 60% a 90%). Ou seja, mau pagadores com pedidos de crédito mais baixos são mais frequentes, porém pedidos maiores de crédito possuem maior chance de não serem pagos. Um ponto de atenção aqui é o tamanho da amostra para valores maiores de crédito, que pode ser muito pequena quando comparada ao tamanho da amostra de pedidos de crédito menores.

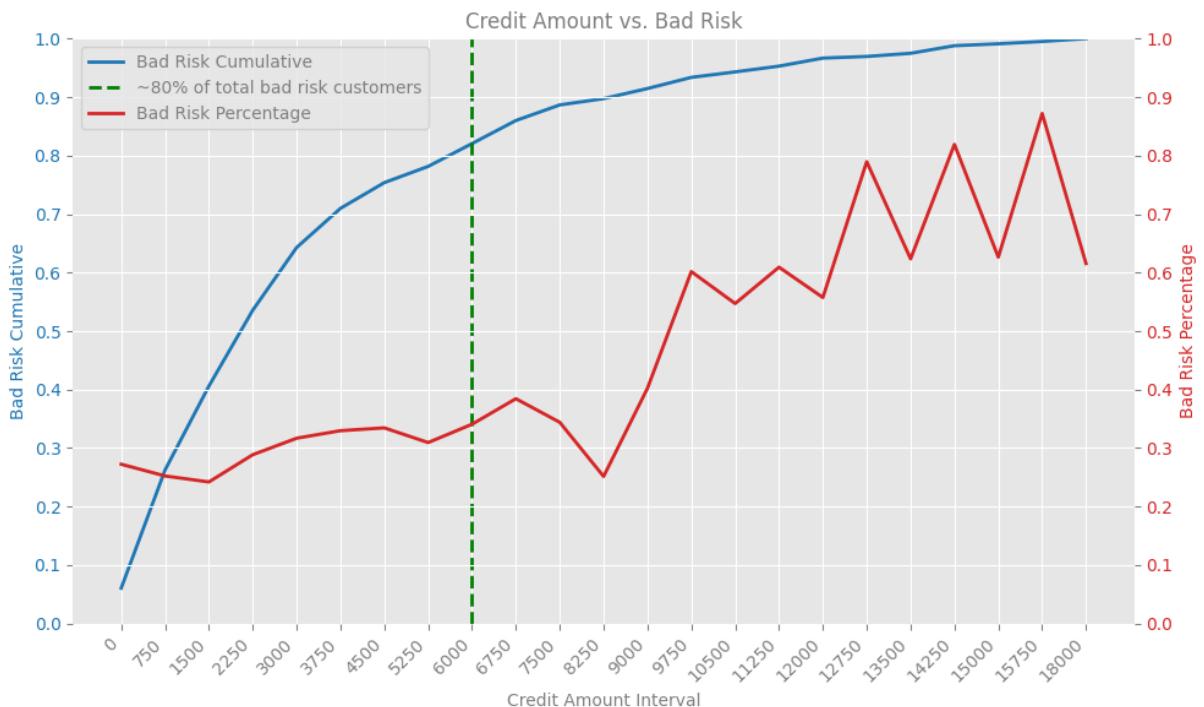


Figura 15 – Comportamento de mau pagadores em diferentes faixas de crédito

Em termos de poupanças, pela figura 16, pode-se observar que clientes com acúmulo financeiro maior tendem a ser menos mau pagadores (aproximadamente 0,18% desses grupos não cumprem com as obrigações financeiras). Porém, eles são bem menos representativos no *dataset*, que conta em sua maioria com pessoas que possuem <100DM em suas contas - e esse grupo possui a maior taxa de mau pagadores.

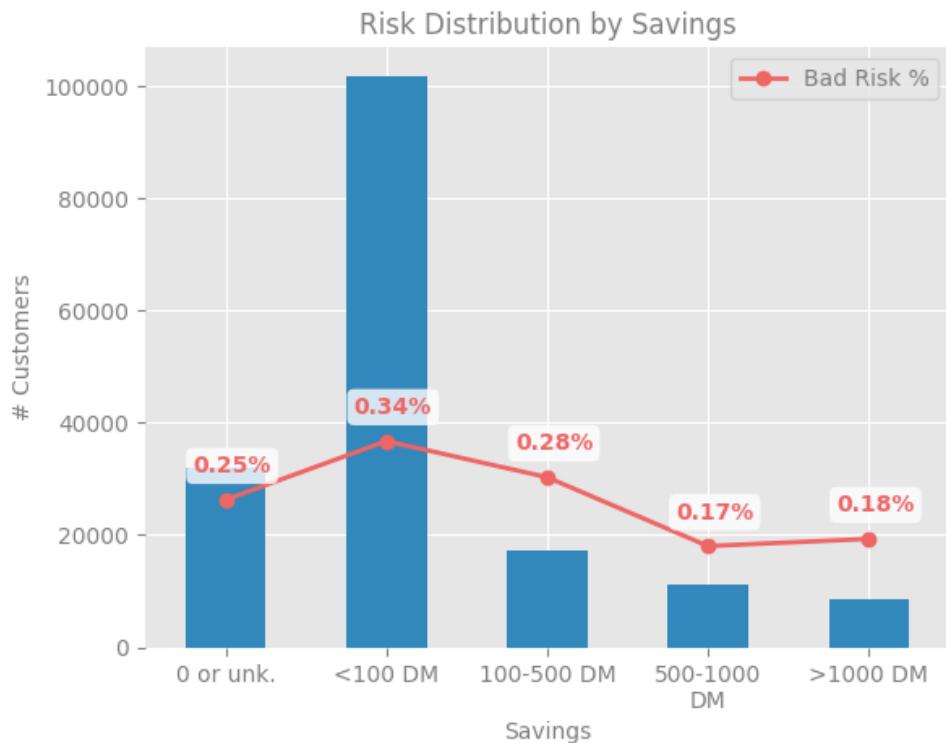


Figura 16 – Distribuição de risco por faixa de poupança

Por fim, outra relação também observada é como o risco varia quando é analisado para diferentes grupos de permanência no emprego atual (isto é, quanto tempo cada cliente está em seu trabalho atual). A Figura 17 indica que há uma pequena diferença entre clientes que estão há menos de 1 ano no emprego vs clientes que estão há mais de 7 anos (0,36% vs 0,25%), o que é coerente com o esperado. Além disso, todos os grupos possuem uma boa representatividade na base, com exceção de desempregados, que são uma amostra relativamente pequena.

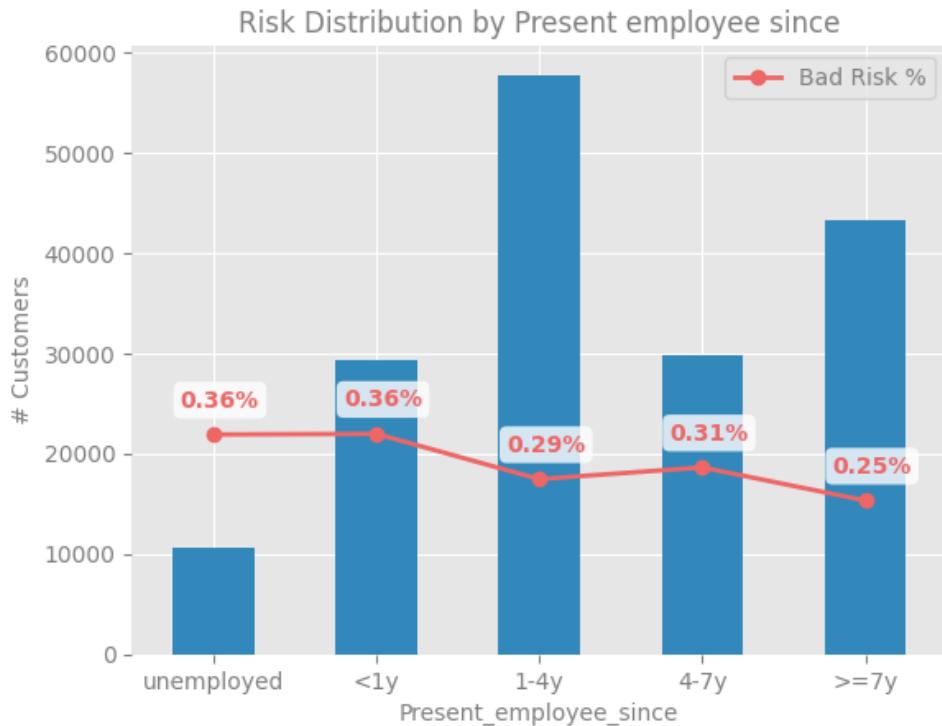


Figura 17 – Distribuição de risco por tempo de permanência no emprego atual

5.2 Métricas

Para poder comparar os algoritmos levantados previamente para análise do risco de crédito bancário, é necessário estabelecer e definir uma métrica a ser comparada. Pode-se então obter a seguinte matriz de confusão:

		Valor obtido	
		Bom pagador	Mau pagador
Valor verdadeiro	Bom pagador	TP	FN
	Mau pagador	FP	TN

Tabela 2 – Matriz de Confusão

Onde:

- TP: o cliente é um bom pagador e foi classificado como bom pagador;
- FP: o cliente é um mau pagador e foi classificado como bom pagador;
- FN: o cliente é um bom pagador e foi classificado como bom pagador;
- TN: o cliente é um mau pagador e foi classificado como um mau pagador.

Pode-se obter as seguintes medidas a partir da matriz de confusão: acurácia, *recall*, especificidade, precisão e F1-score (SUHADOLNIK; UNEYAMA; SILVA, 2023).

A acurácia é uma métrica global de desempenho preditivo que corresponde à proporção de classificações corretas do algoritmo, denotada como:

$$\text{Acurácia} = \frac{\text{TP} + \text{TN}}{\text{N}}$$

Apesar do uso generalizado, a acurácia pode ser insuficiente para lidar com conjuntos de dados desbalanceados que beneficiam uma classe majoritária. A taxa de verdadeiros positivos, também conhecida como *recall* (ou *sensitivity*), é a proporção de casos positivos (bom pagadores) classificados corretamente pelo algoritmo em relação ao número total de verdadeiros positivos:

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

A proporção de casos negativos (mau pagadores) classificados corretamente pelo algoritmo em relação ao número total de negativos genuínos é chamada de especificidade, também conhecida como taxa de verdadeiros negativos:

$$\text{Especificidade} = \frac{\text{TN}}{\text{TN} + \text{FP}}$$

A precisão é a fração de casos positivos (bom pagadores) identificados corretamente pelo algoritmo em relação ao número total de positivos identificados.

$$\text{Precisão} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

O F1-score é calculado como:

$$\text{F1} = \frac{2 \cdot \text{Precisão} \cdot \text{Recall}}{\text{Precisão} + \text{Recall}}$$

Assim, o F1-score fornece uma medida sintética de precisão e sensibilidade superior e mais refinada do que somente a acurácia.

É importante ressaltar que a escolha e priorização de métricas a serem acompanhadas depende diretamente do contexto de negócio e segmentação de clientes. Por exemplo, um algoritmo de análise de crédito para clientes de alta renda deve acompanhar com mais atenção a métrica de recall, a fim de garantir que os bons pagadores não tenham crédito negado (algo que impacta a experiência de clientes alta renda mais do que do varejo).

	Predicted Positive	Predicted Negative	
Actual Positive	TP <i>True Positive</i>	FN <i>False Negative</i>	Sensitivity $\frac{TP}{(TP+FN)}$
Actual Negative	FP <i>False Positive</i>	TN <i>True Negative</i>	Specificity $\frac{TN}{(TN+FP)}$
	Precision $\frac{TP}{(TP+FP)}$	Negative Predictive Value $\frac{TN}{(TN+FN)}$	Accuracy $\frac{TP + TN}{(TP+TN+FP+FN)}$

Figura 18 – Métricas

5.3 Algoritmos

O *dataset* previamente tratado e expandido para um maior conjunto de dados (5.1) foi utilizado para a execução dos algoritmos.

Para todos os algoritmos testados, o *dataset* foi dividido em duas partes: *features* (X) e variável-alvo (Y). Para a variável-alvo, foi definida a coluna "risk", ou seja, ela é a variável que queremos prever (se um determinado cliente é um bom ou mau pagador). Além de separar em *features* e variáveis-alvo, também foi necessário dividir o *dataset* entre dados de treinamento e dados de teste (Tabela 3). Para o propósito desse trabalho, foi utilizado *test_size* = 0.2 e *random_state* = 42, que garantem que a amostra de teste seja 20% do *dataset* original e que a divisão seja sempre a mesma para cada execução do código, garantindo reproduzibilidade.

Paramétrico	Valor
<i>test_size</i>	0.2
<i>random_state</i>	42

Tabela 3 – Parâmetros utilizados

5.3.1 Regressão Logística

A regressão logística (2.1.2.1) é um algoritmo de aprendizado de máquina utilizado para problemas de classificação binária.

O modelo foi treinado utilizando o parâmetro *solver* configurado como *saga* e o número máximo de iterações (*max_iter*) definido como 1000. O otimizador *saga* é particularmente indicado para conjuntos de dados de maior porte, por sua eficiência

e escalabilidade. Além disso, o aumento no número de iterações, em relação ao valor padrão de 100, visa assegurar a convergência do modelo durante o processo de ajuste dos parâmetros.

Os resultados obtidos estão representados na Tabela 4 e na matriz de confusão (Figura 19).

Métrica	Valor
Acurácia	0.7256
Recall	0.2779
Especificidade	0.9175
Precisão	0.7477
F1	0.8240
Tempo de treinamento	20.58s

Tabela 4 – Regressão Logística - Resultados obtidos

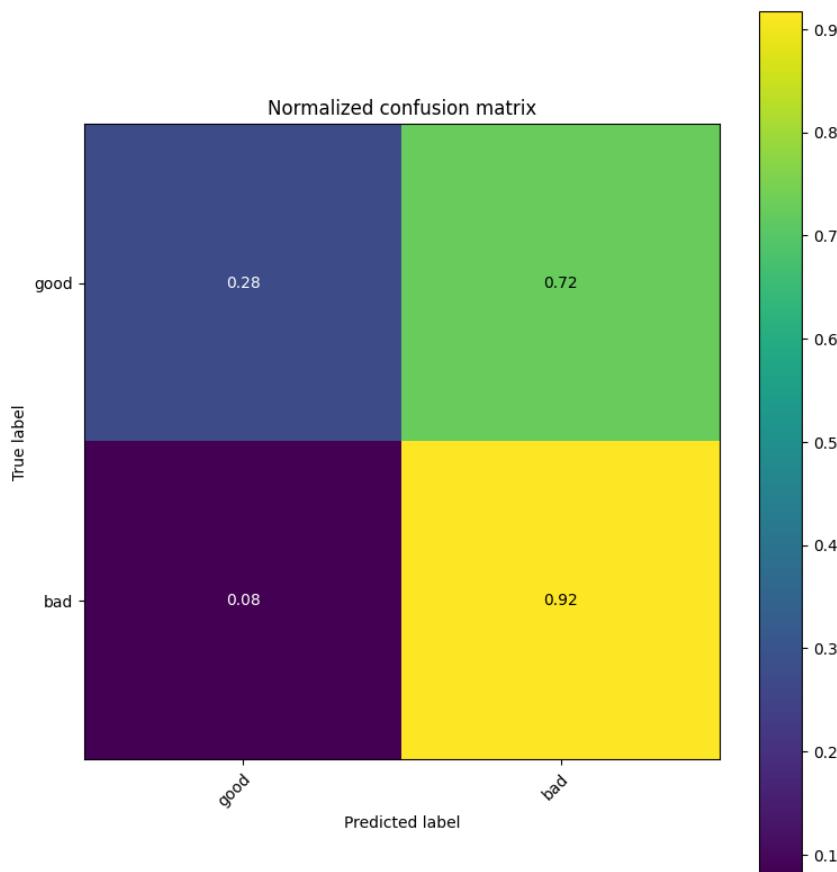


Figura 19 – Regressão Logística - Matriz de confusão

O algoritmo, considerando o conjunto de dados fornecido e os parâmetros avaliados, apresentou uma acurácia de 73%, o que indica que foi capaz de classificar corretamente 73% dos casos analisados. No entanto, é importante destacar que, nesse contexto, os erros de classificação possuem implicações distintas. Classificar um bom pagador como

inadimplente (caso A) representa uma falha que impacta negativamente a experiência do cliente, mas não acarreta prejuízos financeiros diretos à instituição. Por outro lado, classificar um inadimplente como bom pagador (caso B) pode resultar em perdas financeiras significativas. Diante disso, é fundamental analisar separadamente esses dois tipos de erro.

A métrica de *recall* indica que, dentre todos os indivíduos que são bons pagadores, 28% efetivamente obtiveram a concessão de crédito, refletindo a baixa capacidade do modelo em identificar corretamente os casos positivos. A especificidade, por sua vez, avalia a proporção de casos negativos corretamente identificados, revelando que 92% dos indivíduos que não deveriam receber crédito foram, de fato, recusados. Esse resultado evidencia que o modelo apresentou desempenho insatisfatório no caso A, resultando na negação de crédito para bons potenciais pagadores.

Quanto à precisão, observou-se que 75% dos indivíduos que receberam crédito efetivamente atendiam aos critérios para concessão. O modelo alcançou um F1-score de 82% e apresentou tempo de treinamento de 21 segundos.

5.3.2 Random Forest

O *Random Forest*, como já descrito previamente na seção 2.1.2.2, é um algoritmo de aprendizado de máquina que combina diversas árvores de decisão para obter um resultado mais preciso, diminuindo a influência de *overfitting*, por exemplo.

O modelo foi treinado usando $n_estimators = 40$. Isso define o número de árvores de decisão que serão utilizadas.

Os resultados obtidos estão representados na Tabela 5 e na matriz de confusão (Figura 20).

Métrica	Valor
Acurácia	0.9068
Recall	0.8366
Especificidade	0.9369
Precisão	0.9304
F1	0.9336
Tempo de treinamento	73.88s

Tabela 5 – Random Forest - Resultados obtidos

O modelo, ao ser avaliado com base no conjunto de dados fornecido e nos parâmetros definidos, obteve uma acurácia de 91%. Além disso, a métrica de *recall* revelou que, entre os indivíduos bons pagadores, 84% efetivamente receberam a concessão de crédito. Em termos de especificidade, 94% dos indivíduos que não eram aptos para crédito foram corretamente identificados e negados, indicando a eficiência do modelo em discriminar

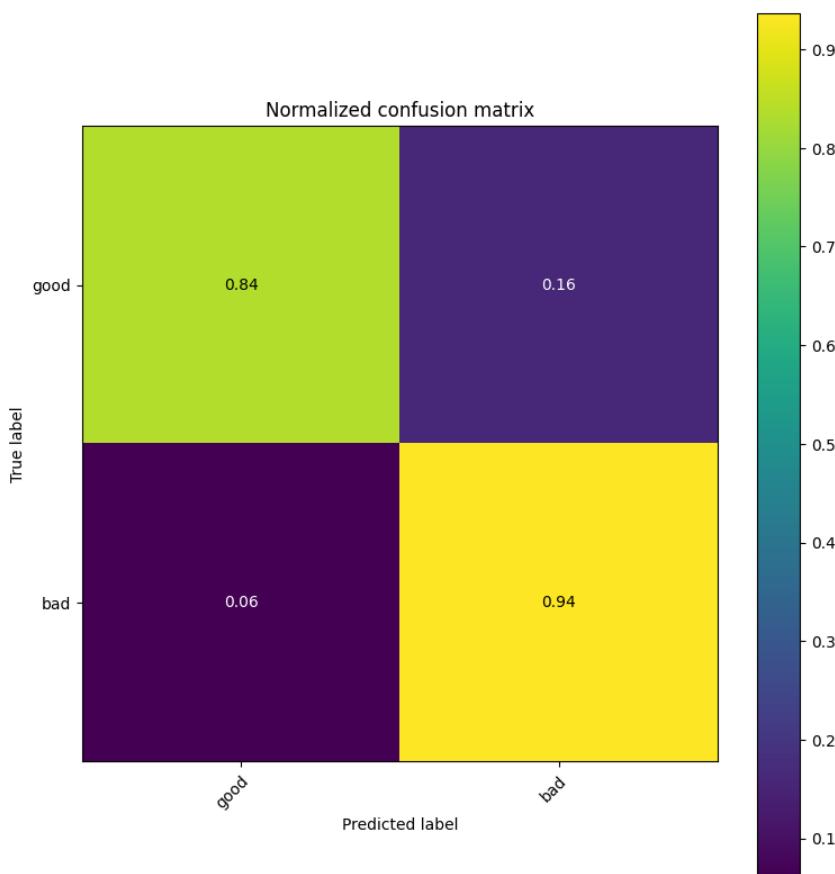


Figura 20 – Random Forest - Matriz de confusão

casos negativos, o que representa uma performance superior à observada na Regressão Logística.

Por fim, a precisão do modelo foi de 93%, evidenciando que a maioria das pessoas que receberam crédito realmente atendiam aos critérios estabelecidos. O modelo obteve um F1-score de 93% e um tempo de treinamento de 74 segundos.

5.3.3 Extreme Gradient Boosting (XGBoost)

Outro algoritmo trabalhado foi o XGBoost, descrito em 2.1.2.3. A principal diferença desse modelo, quando comparado ao Random Forest, está no uso de um treinamento sequencial e aditivo, onde as árvores são construídas uma após a outra (*boosting*), cada uma tentando corrigir os erros da anterior.

Por ser um algoritmo mais sofisticado, ele possui um maior e mais detalhado conjunto de hiperparâmetros que devem ser escolhidos. Para isso, a biblioteca *hyperopt* realiza a busca pelos melhores valores para os hiperparâmetros definidos no espaço de busca. São eles:

- *max depth* (3 a 12): profundidade máxima das árvores, controla a complexidade do

modelo;

- *gamma* (0 a 5): penalização para a criação de novos nós, evita *overfitting*;
- *reg alpha* (0 a 100): regularização L1 (Lasso), adiciona penalidade para pesos elevados e elimina variáveis irrelevantes;
- *reg lambda* (0 a 1): regularização L2 (Ridge), reduz a magnitude dos pesos do modelo e o deixa mais estável/menos sensível a ruídos;
- *colsample bytree* (0.5 a 1): fração de colunas usadas aleatoriamente por árvore, ajuda na diversificação;
- *min child weight* (0 a 10): número mínimo de amostras exigidas em um nó folha, evita divisão excessiva;
- *n estimators* (50 a 300): número total de árvores no modelo, afeta desempenho e *overfitting*;
- *learning rate* (0.01 a 0.3): fator de impacto de cada árvore no resultado final, influencia a velocidade de aprendizado;
- *subsample* (0.5 a 1): fração de amostras usadas em cada árvore, reduz correlação entre árvores e *overfitting*.

Durante o processo de ajuste, a métrica utilizada para avaliar cada combinação de hiperparâmetros foi o F1-score, que é apropriado para problemas de classificação onde há desbalanceamento entre classes ([DEVELOPERS, 2025](#)).

Após encontrar os melhores hiperparâmetros, um novo modelo foi treinado com esses valores e testado no conjunto de teste. O resultado final foi avaliado novamente pelo F1-score, garantindo que a performance fosse otimizada. Todos os hiperparâmetros utilizados e seus respectivos valores finais estão descritos na Tabela 6.

Hiperparamêtro	Melhor valor
<i>colsample_bytree</i>	0.8722
<i>gamma</i>	0.0195
<i>learning_rate</i>	0.1218
<i>max_depth</i>	10
<i>min_child_weight</i>	2
<i>n_estimators</i>	210
<i>reg_alpha</i>	25
<i>reg_lambda</i>	0.5730
<i>subsample</i>	0.7756

Tabela 6 – XGBoost - Parâmetros utilizados

Os resultados obtidos estão descritos na Tabela 7 e ilustrados na Figura 21.

Métrica	Valor
Acurácia	0.9127
Recall	0.8457
Especificidade	0.9414
Precisão	0.9344
F1	0.9378
Tempo de treinamento	37.35s

Tabela 7 – XGBoost - Resultados obtidos

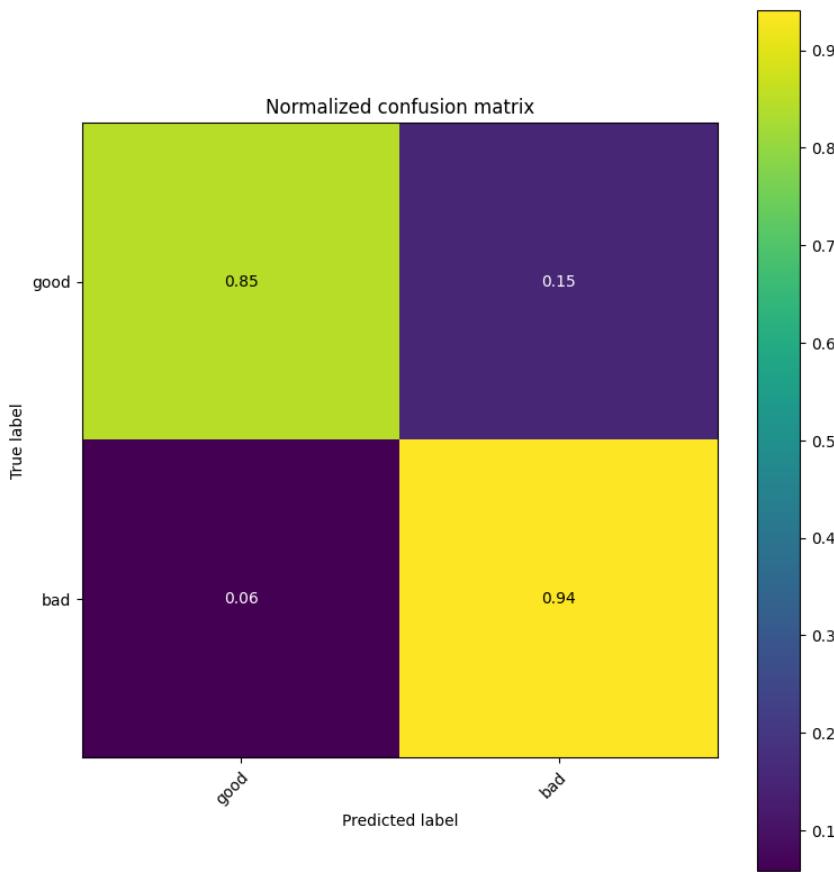


Figura 21 – XGBoost - Matriz de confusão

Em relação a desempenho, a acurácia foi de 91%. A análise da métrica de *recall* indicou que 84% dos bons pagadores foram corretamente contemplados com a concessão de crédito. No que se refere à especificidade, 94% dos indivíduos não elegíveis ao crédito foram devidamente recusados. A precisão, por sua vez, atingiu 93%, demonstrando que a grande maioria dos beneficiários realmente possuía perfil adequado. O F1-score registrado foi de 94% e o tempo necessário para treinamento do modelo foi de 37 segundos. Percebe-se que o XGBoost obteve métricas muito parecidas com as do *Random Forest*, porém com um tempo de treinamento consideravelmente menor.

5.3.4 Multilayer Perceptron (MLP)

O último algoritmo desenvolvido foi o *Multilayer Perceptron* (MLP) (2.1.2.4). O MLP distingue-se dos algoritmos baseados em árvores (*Random Forest* e *XGBoost*) e da Regressão Logística por ser uma rede neural artificial. Essa arquitetura multicamadas permite ao MLP modelar relações não lineares e complexas nos dados de forma mais intrínseca, sem depender da combinação de múltiplos modelos simples, como ocorre nos métodos de *ensemble*. Sua capacidade de aprender representações hierárquicas dos dados o torna especialmente potente para tarefas de classificação complexas, embora com maior custo computacional e necessidade de ajuste de hiperparâmetros.

Para o treinamento do modelo, um conjunto de hiperparâmetros foi ajustado para otimizar seu desempenho e capacidade de generalização. A seguir, são apresentados os principais hiperparâmetros e seus respectivos papéis na configuração da rede neural. Os valores finais atribuídos para cada hiperparâmetro estão na tabela 8.

- *activation*: função de ativação usada entre os neurônios (ex: "relu", "tanh"), responsável por introduzir não-linearidades no modelo;
- *alpha*: termo de regularização L2 que ajuda a evitar *overfitting*, controlando a penalização por pesos grandes;
- *hidden layer size*: define o número e tamanho das camadas ocultas;
- *learning rate init*: valor inicial da taxa de aprendizado usada pelo otimizador para atualizar os pesos durante o treinamento.

Parâmetro	Melhor valor
mlp_activation	relu
mlp_alpha	0.001
mlp_hidden_layer_sizes	(128, 64, 32)
mlp_learning_rate_init	0.01

Tabela 8 – MLP - Parâmetros utilizados

Os resultados obtidos estão descritos na Tabela 9 e ilustrados na Figura 22.

Métrica	Valor
Acurácia	0.9002
Recall	0.8170
Especificidade	0.9359
Precisão	0.9227
F1	0.9292
Tempo de treinamento	783.83s

Tabela 9 – MLP - Resultados obtidos

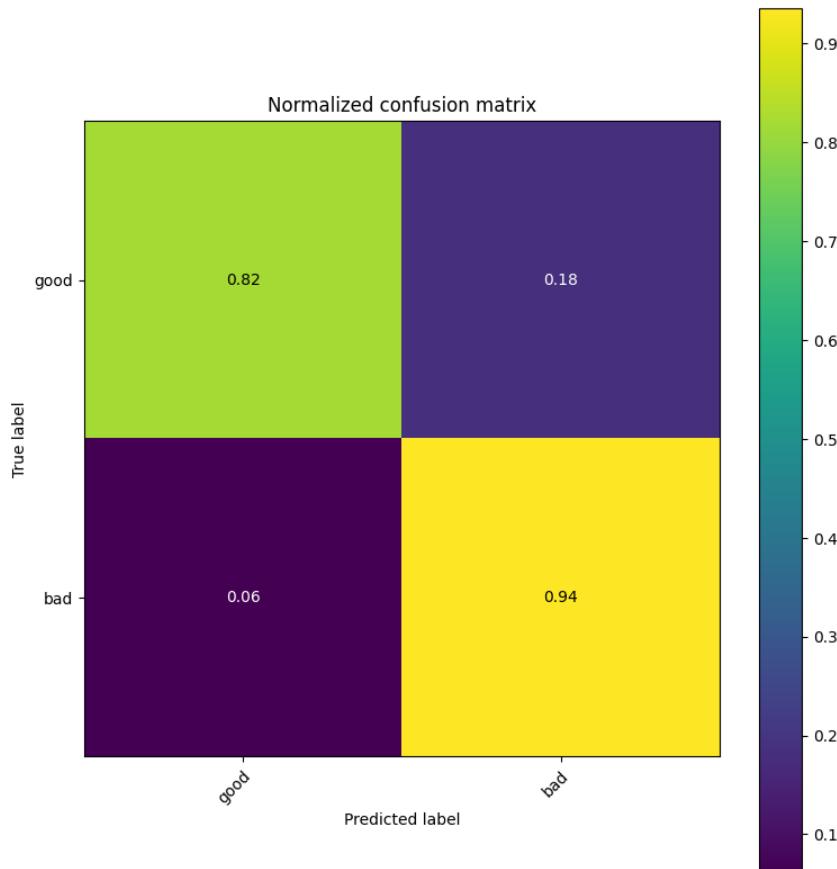


Figura 22 – MLP - Matriz de confusão

Em relação a desempenho, a acurácia avaliada foi de 90%, segundo menor valor entre os algoritmos estudados. A métrica de *recall* mostra que somente 81% dos bons pagadores obtiveram crédito. Para a especificidade, 93% dos indivíduos não elegíveis ao crédito foram devidamente recusados. A precisão foi de 92%, indicando que a maior parte das pessoas que obtiveram crédito realmente merecia. O F1-score registrado foi de 92% e o tempo necessário para o treinamento do modelo foi de 783 segundos. De modo geral, o MLP é mais restrito em relação a concessão de crédito (ponto positivo), porém as custas de um alto tempo de processamento (ponto negativo).

5.4 Análises Comparativas

Para realizar uma comparação entre os resultados obtidos e orientar uma decisão sobre qual algoritmo deveria ser escolhido à luz das necessidades específicas do negócio, foi aplicado o método ATAM (*Architecture Tradeoff Analysis Method*) de maneira simplificada e focada para o contexto em questão. A explicação detalhada do método se encontra em [2.4](#).

5.4.1 Architecture Tradeoff Analysis Method (ATAM)

Baseado na abordagem apresentada por ([CéSAR, 2025](#)), a análise comparativa será realizada em 9 passos principais, detalhados a seguir.

5.4.1.1 Passo 1 – Coleta de Cenários

Neste passo, são definidos os cenários de uso mais críticos, ou seja, as situações operacionais e de negócio que o sistema deve suportar. O objetivo é criar contextos realistas que servirão de referência para avaliar as alternativas arquiteturais frente às exigências do banco.

Definem-se como cenários críticos que o sistema de apoio a decisões de crédito deve suportar:

- Processar diariamente um número elevado de solicitações em um período de tempo delimitado;
- Evitar a concessão indevida de crédito a clientes inadimplentes (minimizar falsos positivos);
- Não negar crédito a bons pagadores (minimizar falsos negativos).

A especificação desses cenários assegura que a avaliação reflita tanto demandas operacionais quanto metas de negócio, fornecendo referências claras para comparação entre as alternativas arquiteturais.

Para o exercício deste trabalho, delimita-se como contexto de negócio a avaliação de crédito bancário do varejo, focado na oferta para pessoas físicas e com perfis financeiros de entrada, excluindo-se clientes de alta renda e alto patrimônio. Importante ressaltar que, caso o contexto de negócio mude, o exercício de priorização de métricas e o ATAM precisa ser refeito e adaptado para o novo cenário.

5.4.1.2 Passo 2 – Coleta de Requisitos, Restrições e Ambiente

Aqui elicitam-se os requisitos não-funcionais (latência, métricas preditivas) e as limitações ambientais (CPU única, memória). O propósito é reunir todas as condições sob as quais os algoritmos serão testados, garantindo transparência e reproduzibilidade dos resultados.

Elicitam-se requisitos não-funcionais e condições de execução, com base nos requisitos definidos no capítulo [4](#) (Especificação):

- Acurácia $\geq 80\%$ (RNF01)

- Tempo de treinamento menor possível (RNF02)
- Infraestrutura: CPU única e memória limitada

5.4.1.3 Passo 3 – Descrição das Arquiteturas

Cada algoritmo é descrito em termos de sua estrutura e fluxo de dados, destacando componentes-chave (tipo de modelo, número de árvores, camadas ocultas). O objetivo é oferecer uma visão clara das diferenças de complexidade, interpretabilidade e capacidade de capturar padrões entre as soluções candidatas.

- Regressão Logística: combina linearmente variáveis de cliente e aplica função sigmoide; inferência muito rápida e fácil de interpretar, mas limitada em fronteiras não lineares;
- *Random Forest*: agrupa votos de 40 árvores de decisão treinadas em amostras *bootstrap*; robusta a ruídos e capaz de capturar relações não lineares médias, com custo computacional moderado.
- XGBoost: aplica *boosting* sequencial em 210 árvores, corrigindo erros residuais e usando regularização L1/L2; alto poder preditivo e controle de *overfitting*, mas requer *tuning* cuidadoso;
- MLP: rede neural feed-forward com três camadas ocultas (128→64→32 neurônios) e ReLU; modela interações complexas, porém com inferência custosa e sensível ao volume de dados. Dentre todas as opções mapeadas, é a que apresenta o maior efeito de "caixa opaca", prejudicando a interpretabilidade e a capacidade de auditoria dos dados.

5.4.1.4 Passo 4 – Análise dos Atributos Específicos

Neste estágio analisam-se isoladamente as métricas de qualidade e o tempo de inferência de cada algoritmo, com base nos testes de desempenho e submetidos às mesmas condições de execução. O propósito é identificar pontos fortes e fracos de cada opção, fornecendo a base quantitativa para comparações posteriores.

Métricas	Regressão Logística	Random Forest	XGBoost	MLP
Acurácia	0,7256	0,9068	0,9127	0,9002
Recall	0,2779	0,8366	0,8457	0,8170
Especificidade	0,9175	0,9369	0,9414	0,9359
Precisão	0,7477	0,9304	0,9344	0,9227
F1	0,8240	0,9336	0,9378	0,9292
Tempo de treinamento	20,58s	73,88s	37,35s	783,83s

Tabela 10 – Resultados obtidos

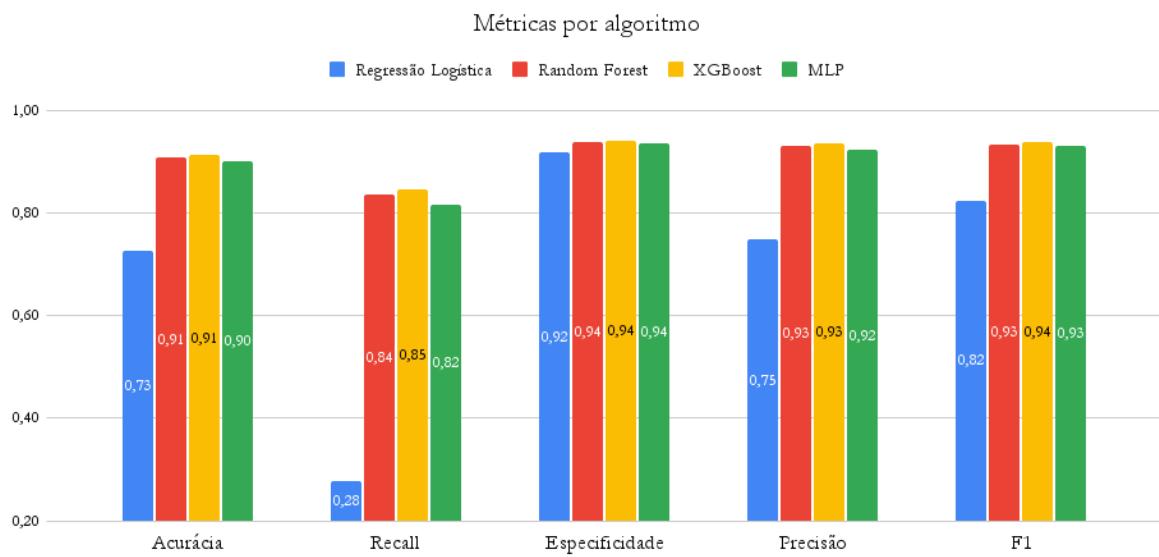


Figura 23 – Métricas dos algoritmos

5.4.1.5 Passo 5 – Identificação de Sensibilidades

Avaliam-se possíveis variações de desempenho sob diferentes condições de ambiente ou parâmetros (volume de dados, número de árvores, etc.).

Nesse caso específico, a avaliação das sensibilidades mostra que o MLP tem tempo de treinamento que cresce exponencialmente com o volume de dados, chegando a quase 800 s em 1 milhão de registros. Tanto o *Random Forest* quanto o XGBoost apresentam sensibilidade moderada ao número de árvores, mas possuem menor tempo de treinamento do que o MLP (aproximadamente 74 s e 37 s, respectivamente). A Regressão Logística, embora ofereça baixo tempo de treinamento, demonstra queda acentuada em *recall*, comprometendo a capacidade de identificar bons pagadores.

5.4.1.6 Passo 6 – Identificação de Trade-offs

Contrastam-se compromissos entre atributos conflitantes. O propósito é documentar as concessões necessárias em cada alternativa, ajudando a balancear múltiplos critérios de decisão de forma fundamentada.

Dentre os *trade-offs*, observou-se que o desempenho computacional contrasta fortemente com a complexidade: a Regressão Logística é treinada em 20 s, seguida por XGBoost (37 s), *Random Forest* (74 s) e MLP (784 s). Em termos de equilíbrio entre *recall* e especificidade, XGBoost ($0,8457 \times 0,9414$) e *Random Forest* ($0,8366 \times 0,9369$) entregam resultados comparáveis, mas o primeiro apresenta leve vantagem no F1-score combinado com tempo de treinamento inferior (XGBoost: 0,9378 em 37 s vs. *Random Forest*: 0,9336 em 74 s).

Esses compromissos evidenciam que nenhuma opção domina em todos os aspectos, sendo necessário ponderar requisitos críticos.

5.4.1.7 Passo 7 – Priorização de Cenários

Neste ponto, *stakeholders* atribuem pesos relativos aos cenários conforme impacto no negócio (segurança versus experiência do usuário). O objetivo é orientar a reanálise para focar nos requisitos mais críticos, assegurando que a escolha final atenda às prioridades estratégicas.

Na priorização hipotética conduzida com *stakeholders*, o critério de minimizar falsos positivos (segurança financeira) foi classificado como mais importante. Essa definição orienta a reanálise, garantindo foco nos requisitos centrais do negócio. Diferentes contextos de negócio e segmentos de clientes podem necessitar de priorizações diferentes. Outro fator extremamente relevante no contexto do mercado bancário é a interpretabilidade do modelo e a capacidade de auditoria dos resultados obtidos.

5.4.1.8 Passo 8 – Reanálise com Cenários Priorizados

Reavaliaram-se as métricas de cada algoritmo sob as prioridades definidas, validando quais alternativas cumprem os requisitos mais importantes. O propósito é confirmar empiricamente a robustez e adequação de cada solução ao contexto real do banco.

Submetidas às prioridades definidas, as métricas confirmam que o XGBoost atende os requisitos de forma mais completa do que o *Random Forest*. As métricas de acurácia, *recall*, especificidade, precisão e F1 são mais altas, além de um tempo de treinamento menor. Um *trade-off* dessa escolha é uma menor capacidade de interpretabilidade. Os outros modelos (Regressão Logística e MLP) ficam aquém nas prioridades definidas.

5.4.1.9 Passo 9 – Apresentação dos Resultados e Recomendação

Consolida-se um relatório final que reúne sensibilidades, *trade-offs* e análises, culminando na recomendação da arquitetura mais equilibrada. O objetivo é fornecer subsídios claros e acionáveis para apoiar a adoção do algoritmo selecionado em produção.

A recomendação destaca as sensibilidades e *trade-offs* identificados e indica a adoção de XGBoost como arquitetura padrão para apoio diário à decisão de crédito. Para mitigar riscos residuais, sugere-se o ajuste fino de hiperparâmetros, assegurando equilíbrio entre alta precisão preditiva e desempenho operacional. Sendo assim, se fosse necessária a escolha de somente um algoritmo, o XGBoost seria o mais adequado.

6 Considerações Finais

6.1 Conclusões

Ao término deste trabalho, que se propôs a realizar uma análise comparativa de algoritmos de *Machine Learning* para a concessão de crédito bancário, foi possível atingir os objetivos centrais delineados. A aplicação dos modelos de Regressão Logística, *Random Forest*, XGBoost e *Multilayer Perceptron* (MLP) sobre um conjunto de dados expandido sinteticamente permitiu não apenas a extração de métricas de desempenho, mas também uma avaliação qualitativa e estruturada por meio do método ATAM (*Architecture Tradeoff Analysis Method*).

A partir das análises, conclui-se que:

- 1. A escolha do modelo é contextual:** A aplicação do método ATAM evidenciou que a seleção de um algoritmo ótimo é intrinsecamente dependente do contexto de negócio. A recomendação do XGBoost, por exemplo, foi derivada de um cenário hipotético focado no varejo, que priorizava a minimização de falsos positivos (concessão a maus pagadores). Fatores como o ambiente regulatório, o cenário macroeconômico, o apetite estratégico a risco da instituição e a maturidade da infraestrutura tecnológica, poderiam alterar drasticamente a ponderação dos atributos de qualidade e, consequentemente, a escolha do modelo.
- 2. A supervisão humana é relevante:** Apesar da elevada acurácia e do poder preditivo demonstrado por modelos como o XGBoost e o *Random Forest*, a implementação de tais sistemas em um ambiente produtivo não prescinde da supervisão humana qualificada. Esta supervisão não se traduz necessariamente em uma validação manual de cada solicitação de crédito, mas sim no estabelecimento de uma malha de controle robusta, com o monitoramento contínuo de Indicadores Chave de Desempenho (KPIs) para garantir que as decisões automatizadas permaneçam alinhadas aos objetivos estratégicos da instituição.
- 3. A abordagem híbrida como alternativa:** A análise evidenciou que a escolha de um único algoritmo "vencedor" não é a única abordagem viável. Sistemas de concessão de crédito podem se beneficiar da implementação de uma abordagem híbrida, onde múltiplos modelos atuam em paralelo. Compreender as características de cada algoritmo, conforme detalhado no ATAM, é fundamental para orquestrar essa colaboração de forma eficaz.

4. **O *trade-off* entre performance e interpretabilidade:** Observou-se um claro *trade-off* entre o desempenho preditivo e a interpretabilidade dos modelos. Enquanto a Regressão Logística oferece resultados facilmente interpretáveis, sua capacidade de modelar relações não lineares é limitada. Em contrapartida, modelos como o MLP e o XGBoost, embora mais performáticos, operam como "caixas opacas", dificultando a explicação de decisões individuais — um requisito cada vez mais presente em ambientes regulados. Com o objetivo de se aprofundar no tema, pode-se consultar o Apêndice A, que aborda o tema de Explainable AI.

6.2 Contribuições

A principal contribuição deste trabalho reside na aplicação de um método formal de análise de *trade-offs* de arquitetura, o ATAM, a um problema de seleção de modelos de *Machine Learning*. Diferentemente de estudos que se limitam à comparação de métricas de performance, este trabalho propõe um *framework* que conecta o desempenho técnico dos algoritmos aos objetivos estratégicos e requisitos não-funcionais do negócio. A especificação do projeto, fundamentada nas visões do RM-ODP, também conferiu uma base robusta e estruturada para o desenvolvimento, representando uma contribuição metodológica para projetos de IA aplicada.

6.3 Perspectivas de Continuidade

Os resultados e a metodologia aqui apresentados abrem caminho para diversas frentes de pesquisa e desenvolvimento futuras:

1. **Aprofundamento da análise por segmento de clientes:** O presente estudo utilizou uma base de dados unificada. Uma continuação natural seria aplicar a mesma metodologia comparativa a segmentos específicos de clientes (e.g., alta renda, não correntistas), investigando se diferentes perfis demandam modelos com características distintas, o que poderia levar a um sistema de múltiplos modelos especializados.
2. **Desenvolvimento de uma interface de controle de sensibilidade:** O limiar de decisão entre "bom" e "mau" pagador foi fixado em 0.5. Um trabalho futuro poderia envolver o desenvolvimento de uma interface que permita ao analista de crédito ou gestor de risco ajustar dinamicamente este limiar (*threshold*). Isso permitiria simular cenários e tornar o sistema uma ferramenta de análise estratégica, e não apenas operacional.
3. **Automação do monitoramento e atualização do modelo:** Um modelo de crédito, uma vez em operação, está sujeito à perda de performance à medida que

o comportamento dos consumidores e as condições de mercado evoluem, fazendo com que ele se torne desatualizado. Um trabalho futuro poderia focar na criação de mecanismos para monitorar o desempenho do modelo de forma contínua. Caso uma queda em sua precisão seja detectada, o sistema poderia sinalizar a necessidade ou até mesmo iniciar um processo de retreinamento automático, utilizando dados mais recentes. Isso garantiria que o modelo se mantenha sempre calibrado com a realidade do mercado, preservando sua eficácia ao longo do tempo.

O acesso ao trabalho desenvolvido até o momento está descrito em detalhes no Apêndice B.

Referências

- ACADEMY, D. S. *Introdução às Redes Adversárias Generativas (GANs – Generative Adversarial Networks)*. 2022. Disponível em: <<https://www.deeplearningbook.com.br/introducao-as-redes-adversarias-generativas-gans-generative-adversarial-networks/>>. Acesso em: 21 jan 25. Citado na página 19.
- ALBUQUERQUE, M. *70% dos trabalhadores brasileiros ganham até dois salários mínimos*. 2022. Disponível em: <<https://www.correiobraziliense.com.br/brasil/2022/11/5053834-70-dos-trabalhadores-brasileiros-ganham-ate-dois-salarios-minimos.html>>. Acesso em: 26 jan 25. Citado na página 13.
- AWS. *What is logistic regression*. 2025. Disponível em: <<https://aws.amazon.com/pt/what-is/logistic-regression/>>. Acesso em: 18 jul 2025. Citado na página 20.
- BRADESCO. *Gerenciamento de riscos*. 2023. Disponível em: <<https://www.bradesco.com.br/informacoes-ao-mercado/relatorios-e-planilhas/gerenciamento-de-riscos/>>. Acesso em: 26 jan 25. Citado na página 13.
- BRASIL, B. C. do. *Estatísticas monetárias e de crédito*. 2023. Disponível em: <<https://www.bcb.gov.br/estatisticas/historicomonetariascredito?ano=2023>>. Acesso em: 26 jan 25. Citado na página 14.
- BRASIL, B. C. do. *Relatório de Economia Bancária 2023*. 2024. Disponível em: <<https://www.bcb.gov.br/publicacoes/relatorioeconomiabancaria>>. Acesso em: 26 jan 25. Citado na página 13.
- BRASIL, B. C. do. *Gestão integrada de riscos, compliance e controles internos da gestão*. 2025. Disponível em: <<https://www.bcb.gov.br/acessoinformacao/riscosecontroles>>. Acesso em: 19 abr 2025. Citado na página 11.
- BRASIL, B. do. *Gerenciamento de riscos*. 2023. Disponível em: <<https://ri.bb.com.br/informacoes-financeiras/relatorio-de-gerenciamento-de-riscos/>>. Acesso em: 26 jan 25. Citado na página 13.
- CHANG, V. et al. Credit risk prediction using machine learning and deep learning: A study on credit card customers. *Risks*, v. 12, n. 11, 2024. ISSN 2227-9091. Disponível em: <<https://www.mdpi.com/2227-9091/12/11/174>>. Citado 2 vezes nas páginas 16 e 32.
- CHINOSI, A. T. M. Bpmn: An introduction to the standard. *Computer Standards & Interfaces*, v. 17, p. 124–134, 2012. Citado na página 25.
- CéSAR, J. *The Architecture Tradeoff Analysis Method*. 2025. Disponível em: <<https://homepages.dcc.ufmg.br/~figueiredo/disciplinas/2013b/reuso/paper-ATAM-Julio.pdf>>. Acesso em: 20 jun 2025. Citado na página 52.
- DEVELOPERS, G. for. *Classificação: precisão, recall, precisão e métricas relacionadas*. 2025. Disponível em: <<https://developers.google.com/machine-learning/crash-course/classification/accuracy-precision-recall?hl=pt-br>>. Acesso em: 13 abr 2025. Citado na página 48.

- GOOGLE. *O que é machine learning (ML)?* 2023. Disponível em: <<https://cloud.google.com/learn/what-is-machine-learning?hl=pt-BR>>. Acesso em: 04 fev 25. Citado na página 17.
- GOOGLE. *O que é machine learning (ML)?* 2023. Disponível em: <<https://cloud.google.com/learn/what-is-machine-learning?hl=pt-BR>>. Acesso em: 04 fev 25. Citado na página 22.
- GOV, A. *Mais da metade dos brasileiros vivem em apenas 387 dos mais de 5 mil municípios.* 2024. Disponível em: <<https://agenciagov.ebc.com.br/noticias/202408/mais-da-metade-dos-brasileiros-vivem-em-apenas-387-dos-mais-de-5-mil-municípios-1>>. Acesso em: 26 jan 25. Citado na página 13.
- HAYASHI, V. T.; ARAKAKI, R.; RUGGIERO, W. V. Okiot: Trade off analysis of smart speaker architecture on open knowledge iot project. *Internet of Things*, v. 12, p. 100310, 2020. ISSN 2542-6605. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S2542660520301414>>. Citado na página 26.
- HOFMANN, H. *Statlog (German Credit Data).* 1994. UCI Machine Learning Repository. Disponível em: <<https://doi.org/10.24432/C5NC77>>. Acesso em: 18 jan 25. Citado na página 37.
- IBGE. *PIB cresce em 2022.* 2023. Disponível em: <<https://agenciadenoticias.ibge.gov.br/agencia-sala-de-imprensa/2013-agencia-de-noticias/releases/36371-pib-cresce-2-9-em-2022-efecha-o-ano-em-r-9-9-trilhoes>>. Citado na página 13.
- IBGE. *Internet foi acessada em 72,5 milhões de domicílios do país em 2023.* 2024. Disponível em: <[https://agenciadenoticias.ibge.gov.br/agencia-noticias/2012-agencia-de-noticias/noticias/41024-internet-foi-acessada-em-72-5-milhoes-de-domiciliros-do-pais-em-2023#:~:text=Nos%20domic%C3%ADlios%20do%20pa%C3%A9s%20em,%2C9%25%20nesse%20mesmo%20per%C3%ADodo](https://agenciadenoticias.ibge.gov.br/agencia-noticias/2012-agencia-de-noticias/noticias/41024-internet-foi-acessada-em-72-5-milhoes-de-domiciliros-do-pais-em-2023#:~:text=Nos%20domic%C3%ADlios%20do%20pa%C3%A9s%20em,%2C9%25%20nesse%20mesmo%20per%C3%ADodo.)>. Acesso em: 26 jan 25. Citado na página 13.
- IBGE. *Mais da metade dos brasileiros vivem em apenas 387 dos mais de 5 mil municípios.* 2024. Disponível em: <<https://agenciadenoticias.ibge.gov.br/agencia-noticias/2012-agencia-de-noticias/noticias/39022-taxa-de-desocupacao-cai-a-7-8-em-2023-menor-patamar-desde-2014>>. Acesso em: 26 jan 25. Citado na página 13.
- IBM. *O que é o XGBoost?* 2024. Disponível em: <<https://www.ibm.com/br-pt/think/topics/xgboost#:~:text=Desenvolvido%20por%20Tianqi%20Chen%20da,Java%2C%20Scala%20e%20Julia1.>> Acesso em: 31 mar 25. Citado na página 22.
- ISO/IEC. *Information technology – Open Distributed Processing.* 2009. Disponível em: <<https://committee.iso.org/sites/jtc1sc7/home/projects/flagship-standards/isoiec-10746.html>>. Acesso em: 02 fev 25. Citado na página 23.
- ITAU. *Gerenciamento de riscos e capital.* 2023. Disponível em: <<https://www.itau.com.br/relacoes-com-investidores/resultados-e-relatorios/documentos-regulatorios/pilar-3/>>. Acesso em: 26 jan 25. Citado na página 13.

- KAZMAN MARK KLEIN, P. C. R. *ATAM:Method for Architecture Evaluation*. 2000. Disponível em: <https://insights.sei.cmu.edu/documents/629/2000_005_001_13706.pdf>. Acesso em: 24 mar 25. Citado na página 26.
- LEARN, S. *Neural network models*. 2025. Disponível em: <https://scikit-learn.org/stable/modules/neural_networks_supervised.html>. Acesso em: 01 mar 25. Citado na página 23.
- LIMA, A. C. P. C. D. MÉtodo de arquitetura atam no desenvolvimento de sistemas. *Revista Sinergia*, v. 17, p. 22–30, 2016. Citado na página 27.
- OMG. *Business Process Model And Notation*. 2011. Disponível em: <<https://www.omg.org/spec/BPMN/2.0>>. Acesso em: 07 fev 25. Citado na página 25.
- PEDREGOSA, F. et al. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, v. 12, p. 2825–2830, 2011. Citado na página 35.
- PUTMAN, J. *Architecting with RM-ODP*. [S.l.]: Prentice Hall, 2000. Citado na página 23.
- RUSSEL, S. *Inteligência Artificial*. [S.l.]: Elsevier, 2013. Citado na página 17.
- SANTANDER. *Gerenciamento de risco*. 2023. Disponível em: <<https://www.santander.com.br/ri/gerenciamento-de-risco>>. Acesso em: 26 jan 25. Citado na página 13.
- SUHADOLNIK, N.; UEYAMA, J.; SILVA, S. D. Machine learning for enhanced credit risk assessment: An empirical approach. *Journal of Risk and Financial Management*, v. 16, n. 12, 2023. ISSN 1911-8074. Disponível em: <<https://www.mdpi.com/1911-8074/16/12/496>>. Citado 2 vezes nas páginas 16 e 42.
- SYSTEMS, S. *Random Forests*. 2007. Disponível em: <https://www.stat.berkeley.edu/~breiman/RandomForests/cc_copyright.htm>. Acesso em: 19 mar 25. Citado na página 21.
- YDATA. 2020. Disponível em: <<https://docs.synthetic.ydata.ai/2.0/>>. Acesso em: 06 fev 25. Citado na página 34.
- YDATA Fabric. 2020. Disponível em: <<https://ydata.ai/products/fabric>>. Acesso em: 06 fev 25. Citado na página 34.

Apêndices

APÊNDICE A – XAI - Explainable AI

A evolução dos algoritmos de *Machine Learning*, conforme explorado neste trabalho, introduziu um paradigma onde o aumento da performance preditiva frequentemente ocorre em detrimento da interpretabilidade. Modelos complexos como o XGBoost e as redes neurais MLP, que demonstraram alta eficácia, operam como "caixas opacas". Seus processos de decisão internos, baseados em milhares de parâmetros e interações não lineares, são intrinsecamente opacos ao entendimento humano. É nesse cenário que emerge o campo da Inteligência Artificial Explicável (*XAI - Explainable AI*), cujo objetivo é desenvolver técnicas e modelos capazes de elucidar e justificar suas previsões de forma compreensível para os *stakeholders*.

No contexto da concessão de crédito, a aplicação de XAI transcende o mero interesse acadêmico, tornando-se uma necessidade estratégica e operacional. Para a instituição financeira, a capacidade de entender por que um modelo tomou uma determinada decisão é fundamental. Analistas de risco necessitam confiar no sistema, gestores precisam compreender os fatores que mais influenciam o risco da carteira para refinar estratégias, e engenheiros de dados podem utilizar essa transparência para depurar e aprimorar o modelo com maior eficiência. Sem essa camada de explicação, o modelo, por mais acurado que seja, permanece como uma ferramenta de confiança limitada, cujo comportamento em cenários atípicos é imprevisível.

A importância da explicabilidade se torna ainda mais crítica sob a ótica externa, envolvendo clientes e órgãos reguladores. Um cliente que tem seu pedido de crédito negado tem o direito de receber uma justificativa plausível para a recusa, uma prática que fortalece a transparência e a relação de confiança com o banco. Mais do que uma boa prática, a capacidade de explicar decisões automatizadas é um imperativo regulatório. A Lei Geral de Proteção de Dados (LGPD) no Brasil, por exemplo, prevê o direito do titular dos dados de solicitar a revisão de decisões tomadas unicamente com base em tratamento automatizado de dados pessoais que afetem seus interesses.

Portanto, a implementação de técnicas de XAI seria um passo subsequente natural e de grande valor para este projeto. Ferramentas como SHAP (*SHapley Additive exPlanations*) ou LIME (*Local Interpretable Model-agnostic Explanations*) podem agregar um grande valor aos modelos desenvolvidos. Abaixo podemos ver um exemplo dessa aplicação dentro do contexto do trabalho envolvendo o SHAP.

A.1 SHAP (SHapley Additive exPlanations)

O SHAP é uma das abordagens mais consolidadas em Inteligência Artificial Explícável (XAI). Ele se fundamenta nos valores de Shapley, conceito oriundo da teoria dos jogos cooperativos desenvolvida por Lloyd S. Shapley, laureado com o Prêmio Nobel de Ciências Econômicas em 2012.

Nessa teoria, cada participante de um jogo contribui de forma distinta para o resultado final, e o valor de Shapley define uma maneira justa de distribuir o ganho total entre eles. Em *Machine Learning*, essa ideia é adaptada considerando cada variável de entrada como um "jogador" e a predição do modelo como o "resultado do jogo".

Assim, o SHAP calcula a contribuição média de cada variável para a predição, indicando quanto cada feature aumenta ou reduz o resultado em relação ao valor médio do modelo. Além de permitir compreender, por exemplo, como renda ou idade influenciam a classificação de um cliente, o método também explica o comportamento global do modelo, destacando as variáveis mais determinantes nas decisões.

A.2 Visualizações SHAP

A biblioteca SHAP, implementada em Python, fornece diferentes tipos de visualizações que auxiliam na interpretação dos resultados de maneira intuitiva. As principais utilizadas neste trabalho foram:

A.2.1 Bar Plot

O gráfico de barras (*plots.bar*) apresenta a importância média absoluta dos valores SHAP, permitindo ranquear as variáveis mais influentes de maneira global. Ele é particularmente útil para obter uma visão geral dos fatores mais determinantes nas decisões do modelo, sem considerar o sentido (positivo ou negativo) da influência.

A.2.2 Summary Plot

O *summary plot* combina a importância e a direção do impacto das variáveis no modelo. Cada ponto do gráfico representa uma instância da base de dados, colorido de acordo com o valor da *feature* (alto ou baixo) e posicionado conforme sua influência na predição.

Esse gráfico permite observar, por exemplo, que determinadas variáveis, quando apresentam valores altos, tendem a aumentar a probabilidade de "bom pagador", enquanto outras reduzem essa probabilidade. Além disso, a densidade dos pontos ao longo do eixo

x indica o grau de variabilidade da contribuição de cada variável, uma forma eficaz de visualizar tanto importância global quanto efeitos locais.

A.2.3 Waterfall Plot

O *waterfall plot* é uma visualização local, voltada à explicação de uma predição individual. Ele parte do valor médio esperado do modelo e mostra, passo a passo, como cada variável específica daquela instância contribui para aumentar ou reduzir a predição final.

Esse tipo de gráfico é particularmente útil para casos explicativos, como justificar por que um determinado cliente teve seu crédito negado.

A.3 Resultados obtidos

A aplicação do SHAP aos quatro modelos desenvolvidos — Regressão Logística, *Random Forest*, XGBoost e MLP — evidenciou tanto diferenças quanto padrões recorrentes na forma como cada algoritmo utiliza as variáveis para tomar decisões. Embora as importâncias globais apresentem certa consistência entre os modelos, as explicações locais destacam nuances próprias de cada arquitetura. Para fins de comparação visual, uma mesma amostra de cliente foi utilizada exclusivamente nos *waterfall plots*, permitindo observar de forma coerente como cada modelo atribui contribuições individuais às variáveis. O perfil dessa amostra é apresentado na Tabela 11.

Feature	Valor
Sexo	Feminino
Estado civil	Divorciado
Idade	58
Número de dependentes	1
Profissão	Trabalhadora não qualificada
Trabalhador estrangeiro	Sim
Tempo no emprego atual	≥ 7 anos
Telefone	Não
Tipo de habitação	Gratuita
Tempo na residência atual	4 anos
Propriedade	Nenhuma / Desconhecida
Conta corrente	< 0 DM
Poupança	< 100 DM
Finalidade do crédito	Carro usado
Histórico de crédito	5 anos
Duração do crédito	48 meses
Valor do crédito	6.416 DM
Fiadores	Nenhum
Outros planos de parcelamento	Banco
Créditos no banco	2
Risco	0

Tabela 11 – Perfil do cliente utilizado para comparação dos waterfall plots nos quatro modelos.

A.3.1 Regressão Logística

A análise do modelo de Regressão Logística revela que as três variáveis com maior importância global são "histórico de crédito", "duração do crédito" e "conta corrente", conforme ilustrado na Figura 24.

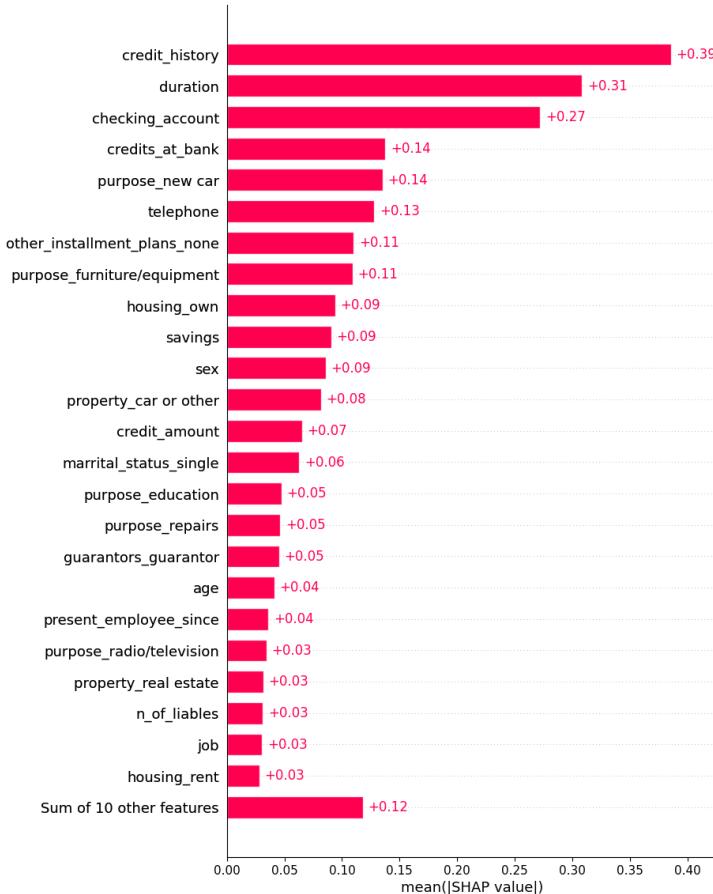


Figura 24 – Importância global das variáveis segundo o modelo Regressão Logística.

No summary plot (Figura 25), observa-se que valores altos de "histórico de crédito" (associados a categorias como "crítico/outros créditos existentes") tendem a aumentar a probabilidade de o cliente ser classificado como "bom pagador", enquanto valores baixos (como "sem histórico/tudo pago") reduzem essa probabilidade. Embora esse padrão pareça contraintuitivo, ele reflete o fato de que clientes sem histórico de crédito — ou com histórico totalmente limpo — são frequentemente considerados mais arriscados por representarem maior incerteza quanto ao comportamento de pagamento. Em relação à "duração" créditos de maior prazo aumentam expressivamente a chance de classificação como "mau pagador", enquanto prazos curtos têm o efeito oposto. Já a "conta corrente" apresenta uma tendência intermediária: saldos muito baixos estão associados a maior probabilidade de ser "bom pagador", enquanto valores medianos tendem a indicar maior risco.



Figura 25 – Summary plot do modelo Regressão Logística.

O *waterfall plot* (Figura 26) foi construído para a amostra da tabela 11. Nesse caso específico, o cliente foi classificado como "mau pagador". O "histórico de crédito" exerceu o maior peso positivo, aumentando a probabilidade de boa classificação; entretanto, as variáveis "duração", "número de créditos no banco" e "conta corrente" apresentaram contribuições negativas que superaram esse efeito, levando à decisão final.

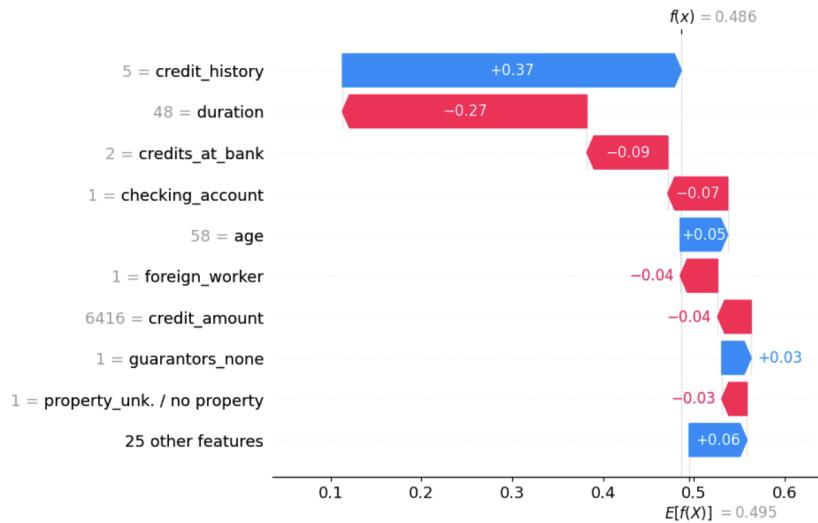


Figura 26 – Waterfall plot do modelo Regressão Logística para a amostra selecionada.

A.3.2 Random Forest

No modelo de *Random Forest*, as variáveis mais relevantes segundo o *bars plot* (Figura 27) foram novamente "conta corrente", "duração" e "histórico de crédito", reforçando a consistência dos resultados entre os modelos.

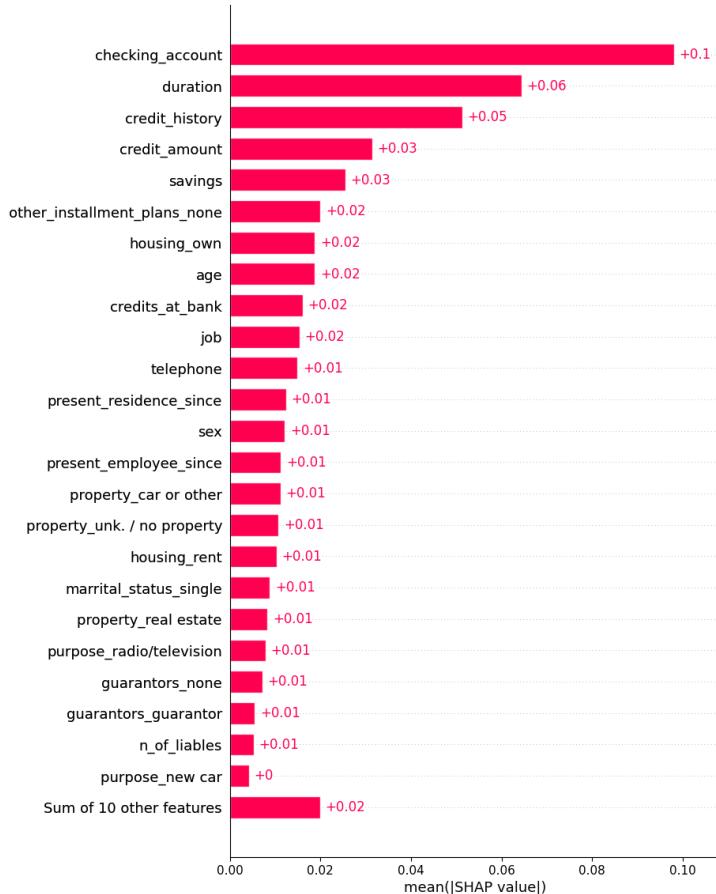
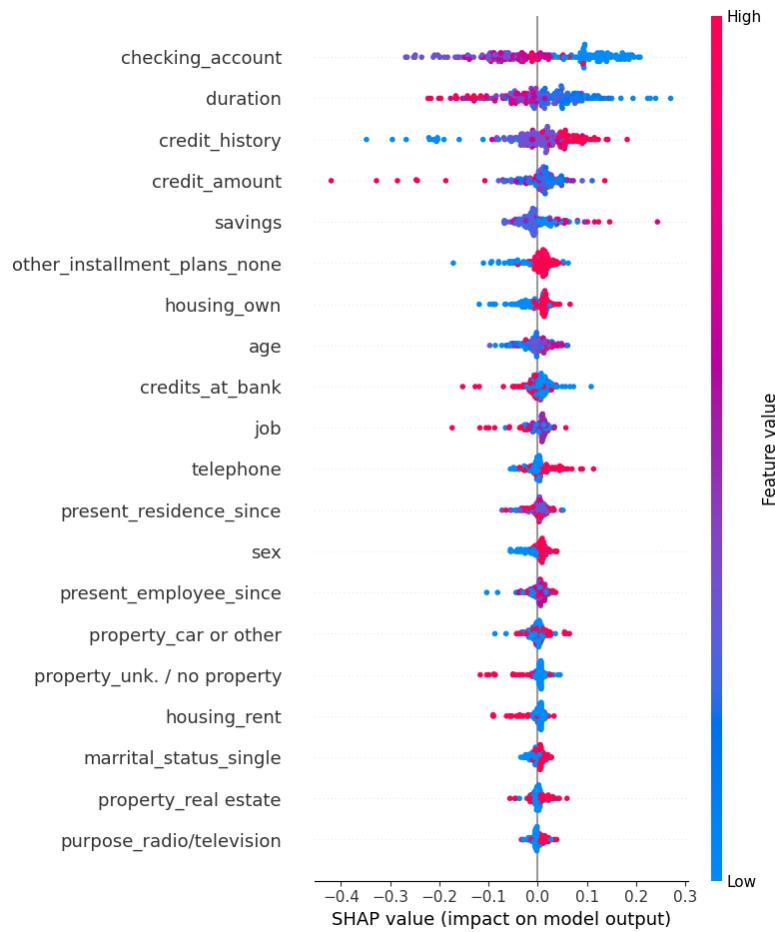


Figura 27 – Importância global das variáveis segundo o modelo *Random Forest*.

O *summary plot* (Figura 28) mostra que valores muito baixos de "conta corrente" (saldo inexistente ou negativo) contribuem fortemente para a classificação como "bom pagador", enquanto valores intermediários aumentam a probabilidade de "mau pagador". A variável "duração" apresenta um padrão semelhante ao observado na regressão logística: prazos longos de crédito associam-se a maior risco, e prazos curtos a menor risco. Já o "histórico de crédito" mostra que clientes com histórico ruim (valores altos) tendem a ser classificados como bons pagadores, enquanto históricos muito bons contribuem de forma oposta.

Figura 28 – Summary plot do modelo *Random Forest*.

O *waterfall plot* (Figura 29) indica que praticamente todas as variáveis atuaram no sentido de reforçar a classificação como "mau pagador", com destaque para as contribuições negativas de "duração", "conta corrente" e "posse de imóvel" (housing).

Figura 29 – Waterfall plot do modelo *Random Forest* para a amostra selecionada.

A.3.3 XGBoost

No modelo XGBoost, as três variáveis de maior importância global são novamente "conta corrente", "duração" e "histórico de crédito", como mostra a Figura 30.

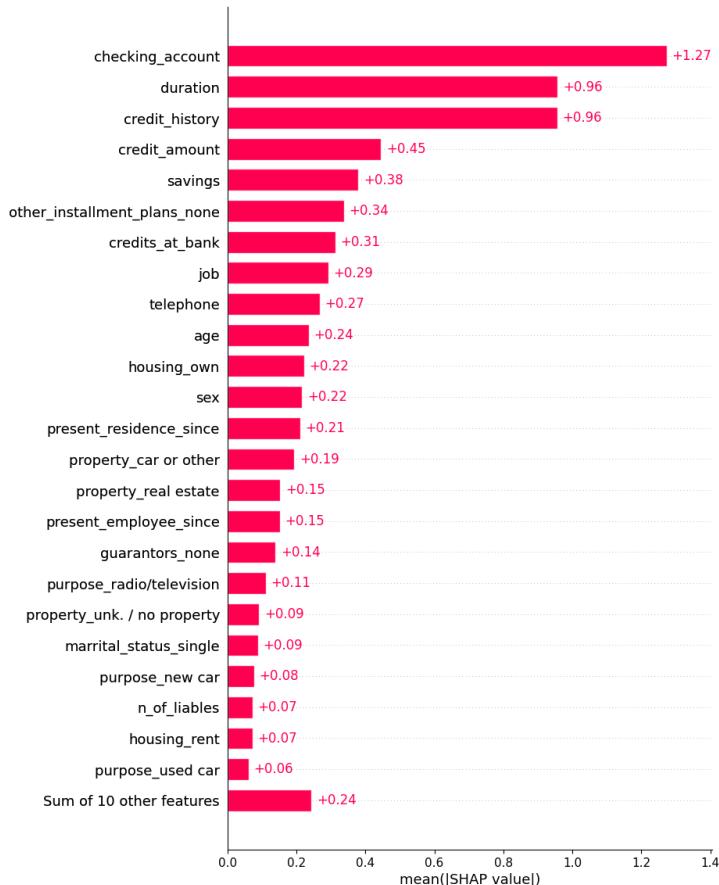


Figura 30 – Importância global das variáveis segundo o modelo XGBoost.

O *summary plot* (Figura 31) evidencia o mesmo padrão observado nos modelos anteriores: "conta corrente" com baixos valores tende a indicar "bom pagador", enquanto valores médios associam-se a "mau pagador". A "duração" segue uma tendência clara — quanto maior o prazo, maior a probabilidade de inadimplência. O "histórico de crédito" mantém o comportamento já identificado: clientes com histórico muito bom tendem a ser classificados como maus pagadores, e históricos piores contribuem moderadamente para o lado de bons pagadores.

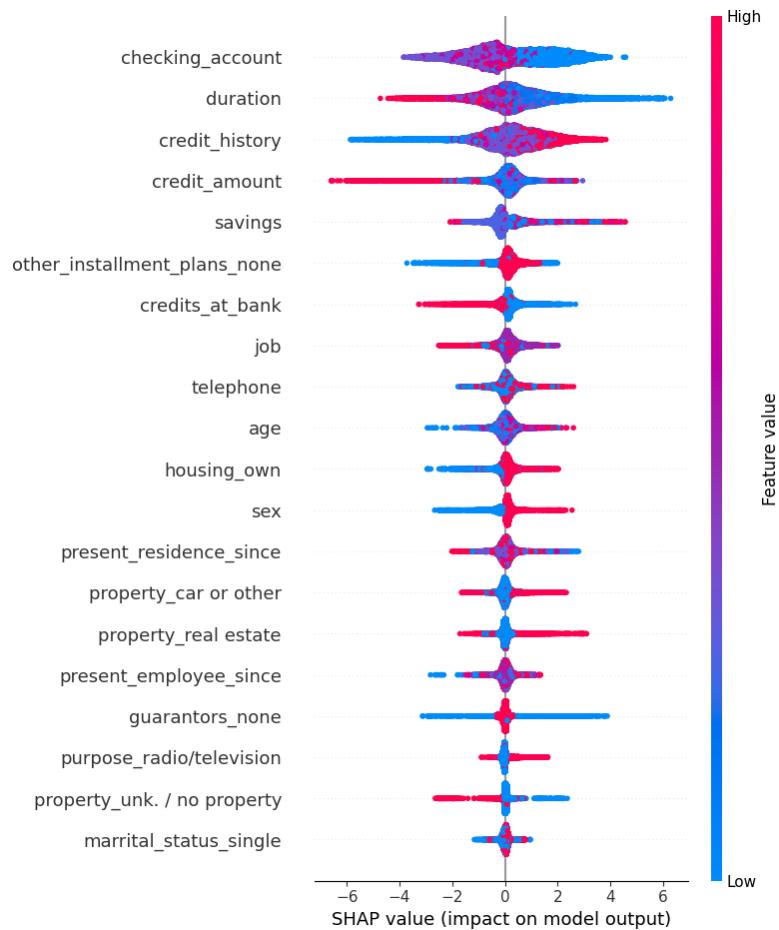


Figura 31 – Summary plot do modelo XGBoost.

No *waterfall plot* (Figura 32), referente à mesma amostra analisada nos demais modelos, observa-se que todas as features contribuem para a classificação final como "mau pagador", com destaque para as variáveis "duração", "sexo" e "telefone", que exerceram as maiores influências negativas.

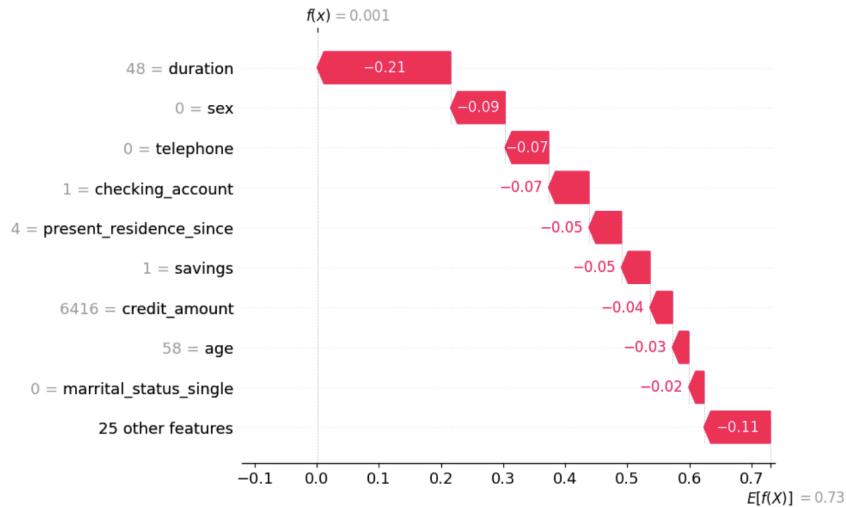


Figura 32 – Waterfall plot do modelo XGBoost para a amostra selecionada.

A.3.4 Multilayer Perceptron (MLP)

Por fim, o modelo MLP apresentou padrão semelhante aos demais, com "conta corrente", "duração" e "histórico de crédito" novamente figurando entre as variáveis de maior importância global (Figura 33).

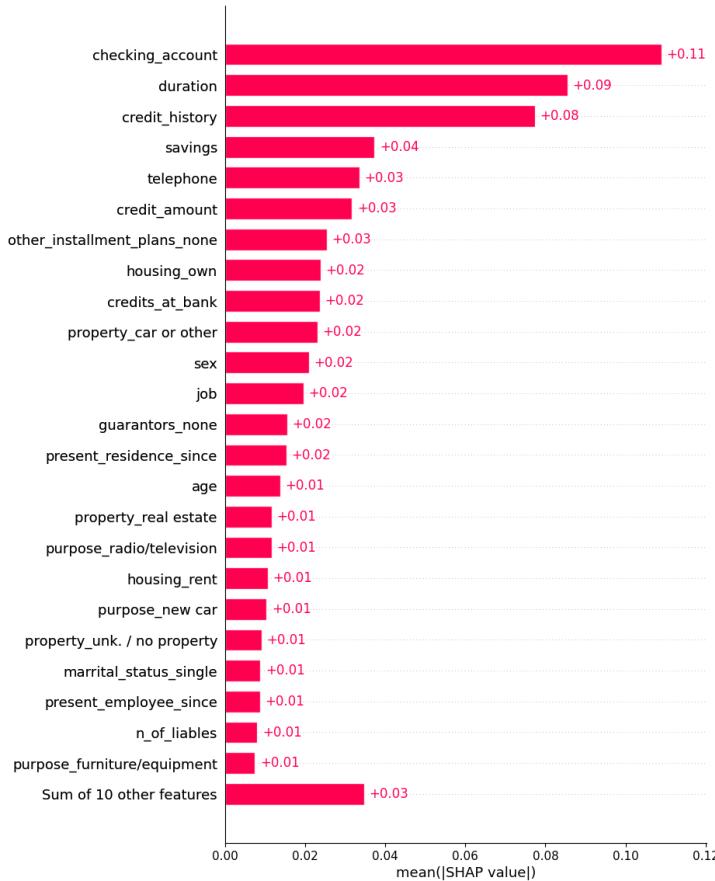


Figura 33 – Importância global das variáveis segundo o modelo MLP.

No *summary plot* (Figura 34), verifica-se que "conta corrente" apresenta o mesmo comportamento observado nos modelos anteriores: valores baixos associam-se a maior chance de "bom pagador", valores intermediários a "mau pagador" e valores altos não apresentam tendência clara. A variável "duração" mantém forte correlação positiva com a inadimplência — prazos longos aumentam o risco, prazos curtos reduzem. Já o "histórico de crédito" segue o padrão discutido anteriormente.

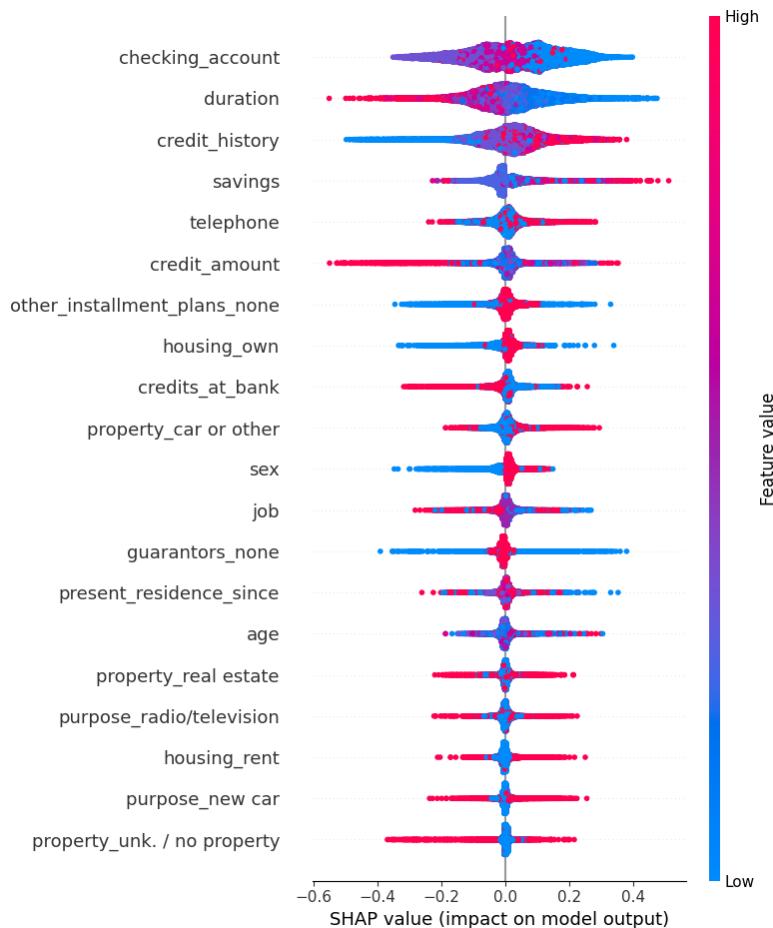


Figura 34 – Summary plot do modelo MLP.

O *waterfall plot* (Figura 35), também com base na mesma amostra utilizada nos outros modelos, mostra que a maior parte das variáveis contribuiu para a classificação como "mau pagador", destacando-se "duração", "número de créditos no banco" e "tempo de residência atual" como as de maior influência.

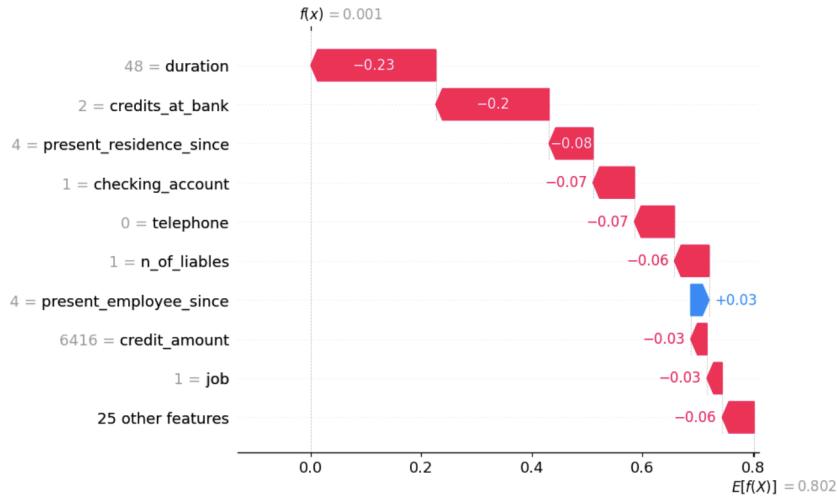


Figura 35 – Waterfall plot do modelo MLP para a amostra selecionada.

A.4 Conclusão

A aplicação de técnicas de XAI, em específico o SHAP, permitiu compreender de forma detalhada o funcionamento interno dos modelos de crédito utilizados neste trabalho. Foi possível identificar padrões consistentes entre os quatro algoritmos avaliados, destacando-se a importância das variáveis "conta corrente", "duração do crédito" e "histórico de crédito" na determinação do risco de inadimplência. Além disso, a análise local por meio dos *waterfall plots* forneceu justificativas claras para decisões individuais, reforçando a transparência e a interpretabilidade do sistema.

Esses resultados demonstram que, mesmo modelos complexos de alto desempenho, como *Random Forest*, XGBoost e MLP, podem ser explicados de maneira comprehensível para *stakeholders* técnicos e não técnicos. A incorporação de XAI não apenas aumenta a confiança na tomada de decisão automatizada, como também atende a requisitos regulatórios e promove uma gestão de risco mais robusta e informada.

APÊNDICE B – Repositório e Reprodutibilidade dos Resultados

Com o objetivo de garantir a transparência e reproduzibilidade deste trabalho, foi desenvolvido um repositório público contendo todos os artefatos relacionados à implementação, treinamento e análise dos modelos, bem como a plataforma web criada para a demonstração dos resultados.

O repositório pode ser acessado em: <<https://github.com/yurialvizi/tcc>>.

A estrutura do repositório é organizada da seguinte forma:

- **models/** – Contém os *Jupyter Notebooks* utilizados para o tratamento dos dados, treinamento dos modelos e análise dos resultados. Dentro dessa pasta há um arquivo **README.md** com instruções detalhadas para a execução dos modelos.
- **frontend/** e **backend/** – Correspondem à plataforma web desenvolvida para apresentar e explicar os resultados obtidos. Essa plataforma permite visualizar as métricas comparativas, as explicações dos modelos (via *SHAP plots*) e utilizar o simulador, no qual é possível inserir dados de um cliente e observar como cada modelo realiza a classificação e a justificativa de sua decisão.

O sistema completo também está disponível online em <<https://tcc-grupo-c06-pcs2025.vercel.app/>>.

Caso o leitor deseje executá-lo localmente, cada pasta possui um arquivo **README.md** com as etapas necessárias para instalação e execução.

Anexos

ANEXO A – Descrição das colunas do dataset original

Description of the German credit dataset.

1. Title: German Credit data

2. Source Information

Professor Dr. Hans Hofmann
Institut f"ur Statistik und "Okonometrie
Universit"at Hamburg
FB Wirtschaftswissenschaften
Von-Melle-Park 5
2000 Hamburg 13

3. Number of Instances: 1000

Two datasets are provided. the original dataset, in the form provided by Prof. Hofmann, contains categorical/symbolic attributes and is in the file "german.data".

For algorithms that need numerical attributes, Strathclyde University produced the file "german.data-numeric". This file has been edited and several indicator variables added to make it suitable for algorithms which cannot cope with categorical variables. Several attributes that are ordered categorical (such as attribute 17) have been coded as integer. This was the form used by StatLog.

6. Number of Attributes german: 20 (7 numerical, 13 categorical)
Number of Attributes german.numer: 24 (24 numerical)

7. Attribute description for german

Attribute 1: (qualitative)
Status of existing checking account
A11 : ... < 0 DM
A12 : 0 <= ... < 200 DM
A13 : ... >= 200 DM /
salary assignments for at least 1 year
A14 : no checking account

Attribute 2: (numerical)
Duration in month

Attribute 3: (qualitative)
Credit history
A30 : no credits taken/
all credits paid back duly
A31 : all credits at this bank paid back duly
A32 : existing credits paid back duly till now
A33 : delay in paying off in the past
A34 : critical account/
other credits existing (not at this bank)

Attribute 4: (qualitative)
Purpose
A40 : car (new)
A41 : car (used)

A42 : furniture/equipment
A43 : radio/television
A44 : domestic appliances
A45 : repairs
A46 : education
A47 : (vacation - does not exist?)
A48 : retraining
A49 : business
A410 : others

Attribute 5: (numerical)
Credit amount

Attribute 6: (qualitative)

Savings account/bonds
A61 : ... < 100 DM
A62 : 100 <= ... < 500 DM
A63 : 500 <= ... < 1000 DM
A64 : ... >= 1000 DM
A65 : unknown/ no savings account

Attribute 7: (qualitative)

Present employment since
A71 : unemployed
A72 : ... < 1 year
A73 : 1 <= ... < 4 years
A74 : 4 <= ... < 7 years
A75 : ... >= 7 years

Attribute 8: (numerical)

Installment rate in percentage of disposable income

Attribute 9: (qualitative)

Personal status and sex
A91 : male : divorced/separated
A92 : female : divorced/separated/married
A93 : male : single
A94 : male : married/widowed
A95 : female : single

Attribute 10: (qualitative)

Other debtors / guarantors
A101 : none
A102 : co-applicant
A103 : guarantor

Attribute 11: (numerical)

Present residence since

Attribute 12: (qualitative)

Property
A121 : real estate
A122 : if not A121 : building society savings agreement/
life insurance
A123 : if not A121/A122 : car or other, not in attribute 6
A124 : unknown / no property

Attribute 13: (numerical)

Age in years

Attribute 14: (qualitative)
Other installment plans
A141 : bank
A142 : stores
A143 : none

Attribute 15: (qualitative)
Housing
A151 : rent
A152 : own
A153 : for free

Attribute 16: (numerical)
Number of existing credits at this bank

Attribute 17: (qualitative)
Job
A171 : unemployed/ unskilled - non-resident
A172 : unskilled - resident
A173 : skilled employee / official
A174 : management/ self-employed/
highly qualified employee/ officer

Attribute 18: (numerical)
Number of people being liable to provide maintenance for

Attribute 19: (qualitative)
Telephone
A191 : none
A192 : yes, registered under the customers name

Attribute 20: (qualitative)
foreign worker
A201 : yes
A202 : no

8. Cost Matrix

This dataset requires use of a cost matrix (see below)

	1	2
1	0	1
2	5	0

(1 = Good, 2 = Bad)

the rows represent the actual classification and the columns
the predicted classification.

It is worse to class a customer as good when they are bad (5),
than it is to class a customer as bad when they are good (1).

