CEFET-MG – Campus Divinópolis
Engenharia de Computação

Relatório Técnico: Algoritmos de Aprendizagem e Otimização

Inteligência Artificial – 2025/2

Rafael Adolfo Silva Ferreira
21 de Dezembro de 2025

Resumo: Este relatório documenta a implementação de soluções de IA integrando abordagens clássicas e bio-inspiradas. Inicialmente, apresenta-se uma **árvore de decisão manual composta por 10 perguntas binárias**, explorando a modelagem de lógica heurística. Na sequência, utiliza-se o dataset **Olist** para investigar a previsibilidade de atrasos logísticos, onde a análise exploratória (EDA) revelou um cenário dominado por ruído externo, limitando modelos clássicos a um teto de acurácia de 60%. O estudo culmina na aplicação de algoritmos de **Computação Natural (GA, PSO e CLONALG)** para a otimização de hiperparâmetros do SVM, confirmando que, embora tais técnicas sejam robustas na busca global, os resultados convergem para os limites práticos e estocásticos observados pela comunidade técnica no Kaggle.

Parte 1: "Escolha seu Sofrimento" – Árvore de Decisão Manual

A primeira etapa do trabalho consistiu na implementação de uma lógica de classificação heurística desenvolvida totalmente "do zero", sem a utilização de bibliotecas de aprendizado de máquina como o **scikit-learn**. O tema escolhido, intitulado "Escolha seu Sofrimento", foca na orientação de carreira técnica por meio da seleção de linguagens de programação e stacks tecnológicas.

A árvore foi estruturada com **10 perguntas binárias** que exploram o perfil do usuário, avaliando critérios como a tolerância à complexidade técnica (gerenciamento manual de memória) e os objetivos profissionais de longo prazo. O fluxo lógico busca separar perfis que priorizam a abstração e produtividade daqueles voltados ao desenvolvimento de sistemas de baixo nível.

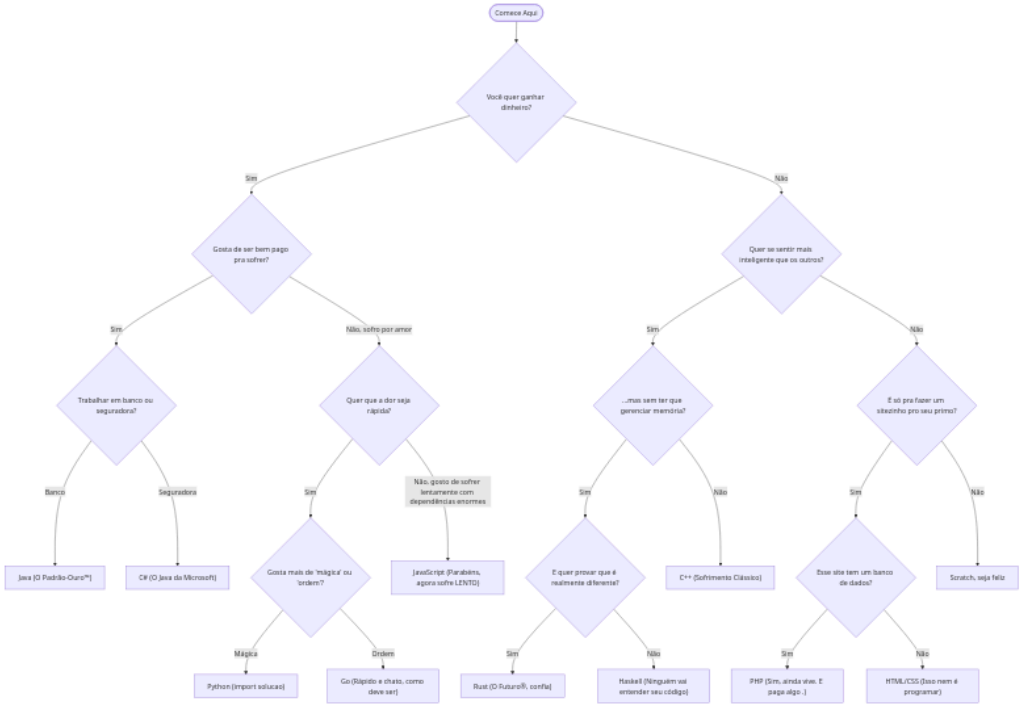

Figura 1: Diagrama manual: uma jornada do "import solution" de Python ao sofrimento clássico do C++[cite: 1].

Exemplos de Execução

Conforme exigido pelas diretrizes técnicas , o script `tree_manual.py` foi validado por meio de execuções interativas no terminal. Abaixo, apresentam-se dois cenários reais de classificação:

Cenário A (Foco em Alta Abstração):

- **Entradas:** Respostas positivas para trabalho em equipe e conforto com tecnologias modernas; resposta negativa para lidar com números puros[cite: 2, 3].
- **Resultado:** A árvore conduz o usuário a recomendações de stacks de alto nível, sugerindo linguagens como **Python ou TypeScript**.

Cenário B (Foco em Baixo Nível/Sistemas):

- **Entradas:** Respostas positivas para resolução de problemas complexos, desafios físicos (lógica de hardware) e interesse em sistemas organizado.
- **Resultado:** O fluxo termina no "sofrimento clássico", recomendando o aprendizado de **C++ ou Engenharia de Sistemas**.

Parte 2: Aprendizado Supervisionado e Engenharia de Dados

A segunda etapa do trabalho focou na aplicação de algoritmos de classificação para prever atrasos logísticos utilizando o **Brazilian E-Commerce Public Dataset by Olist**. O objetivo foi identificar se um pedido seria entregue após o prazo estimado (`is_late = 1`), caracterizando uma tarefa de **classificação binária**.

Estrutura e Atributos do Dataset

O conjunto de dados da Olist consiste em uma base pública de comércio eletrônico brasileiro que reúne informações reais de aproximadamente 100 mil pedidos realizados entre 2016 e 2018.

O diferencial desta base é sua arquitetura relacional composta por 8 tabelas principais, que permitem rastrear o ciclo de vida completo de uma compra:

- **Pedidos (`orders`):** Tabela central que contém o status do pedido e os timestamps vitais para o cálculo de logística.
- **Itens (`order_items`):** Contém dados sobre o preço, valor do frete e a associação entre produtos e vendedores.
- **Produtos (`products`):** Atributos físicos como peso e dimensões, além da categoria do item.
- **Geolocalização:** Informações de localização de clientes e vendedores baseadas em códigos postais.

A variável alvo deste estudo, `is_late`, é um atributo derivado (feature engineering) definido pela comparação entre o prazo estimado de entrega (`order_estimated_delivery_date`) e a data em que o cliente efetivamente recebeu o produto (`order_delivered_customer_date`). Essa modelagem transforma o problema em uma classificação binária, onde o objetivo é prever a falha no cumprimento do prazo prometido.

Estratégia de Dados e Pré-processamento

A análise exploratória (EDA) feita no notebook `EDA.ipynb` revelou um desbalanceamento severo (apenas $\approx 10\%$ de atrasos) e uma correlação fraca entre atributos físicos e o alvo. Para mitigar o viés indutivo e permitir que os modelos aprendessem os sinais de atraso, aplicou-se o seguinte fluxo:

- **Tratamento:** Limpeza de valores nulos e codificação de variáveis categóricas via **One-Hot Encoding**.
- **Escalonamento:** Aplicação de **StandardScaler** para normalizar os dados antes do treino de modelos sensíveis à escala, como KNN e SVM.
- **Balanceamento:** Utilizou-se **Undersampling (50/50)**, selecionando todos os casos de atraso e uma amostra equivalente de casos no prazo.
- **Validação:** Aplicação de **Stratified k-fold (k=5)** para garantir a robustez estatística dos resultados[cite: 23].

Resultados e Tabela Comparativa

Os modelos foram avaliados utilizando as métricas exigidas pelo protocolo experimental. Nota-se que, embora a acurácia global se estabilize, houve um ganho significativo na revocação (**Recall**) após o balanceamento.

Modelo	Acurácia	Precisão	Revocação	F1-Score
KNN (k=5)	57.00%	0.57	0.58	0.5700
SVM (RBF)	59.87%	0.60	0.61	0.5975
Árvore de Decisão	60.13%	0.60	0.62	0.6009

Tabela 1: Comparativo de métricas: estabilização na barreira teórica dos 60%.

Discussão dos Resultados

A convergência dos resultados para a faixa de 60% evidencia um **teto de previsibilidade** inerente à base de dados. Conforme discutido na comunidade Kaggle do Dataset, variáveis críticas como greves dos Correios, condições climáticas e gargalos de infraestrutura não estão mapeadas nos atributos disponíveis. Sob a ótica teórica de Russell & Norvig [1], o ambiente é

parcialmente observável e ruidoso; a ausência de variáveis latentes impede que modelos mais complexos superem a variância estocástica dos dados, tornando o desempenho obtido o limite prático de informação contida no dataset.

Parte 3: Otimização Meta-heurística (AG)

A terceira etapa consistiu na implementação de um Algoritmo Genético (AG) desenvolvido do zero para otimizar os hiperparâmetros C (penalidade de erro) e γ (coeficiente do kernel RBF) do classificador SVM. O objetivo foi verificar se uma busca global e estocástica seria capaz de superar as métricas obtidas na fase de **baseline** e romper o teto de desempenho observado anteriormente.

Modelagem do Algoritmo

A implementação seguiu os critérios de busca em espaços não-convexos, utilizando a seguinte modelagem técnica:

- **Representação (Codificação):** Utilizou-se **Codificação Real** (vetor de ponto flutuante), onde cada indivíduo representa um par (C, γ). O intervalo de busca foi definido entre $[0.1, 100]$ para C e $[0.0001, 1.0]$ para γ.
- **Função de Aptidão (Fitness):** A aptidão é definida pela acurácia média do SVM em um conjunto de validação balanceado. Para viabilizar o processo, utilizou-se uma amostra reduzida de 2.000 instâncias, fundamentada em dois pilares:
  1. **Complexidade Assintótica:** Dado que o custo de treinamento do SVM escala de forma quadrática ou cúbica em relação ao número de amostras ($O(n^2)$ a $O(n^3)$), o uso da base total em cada iteração do AG exigiria um tempo de processamento proibitivo.
  2. **Proxy de Desempenho:** Com base na EDA da Parte 2, uma amostra de 2.000 instâncias balanceadas atua como um **proxy** estatisticamente representativo. Ela permite que o algoritmo identifique a "paisagem" do espaço de busca e a direção do gradiente de melhoria com um erro de estimativa controlado.
- **Operadores Genéticos:** - **Seleção:** Por torneio, garantindo a pressão seletiva necessária.
  - **Crossover:** Aritmético (média ponderada), permitindo a exploração de valores intermediários.
  - **Mutação:** Gaussiana, adicionando um ruído controlado para evitar a estagnação em ótimos locais.
  - **Elitismo:** Preservação dos 2 melhores indivíduos de cada geração (Top-2).

Resultados e Convergência

O algoritmo demonstrou uma convergência precoce, atingindo a estabilidade já na 3ª geração. Esse comportamento, reforçado pela análise de dados prévia, indica que as fronteiras de decisão possíveis em uma base ruidosa são mapeadas rapidamente.

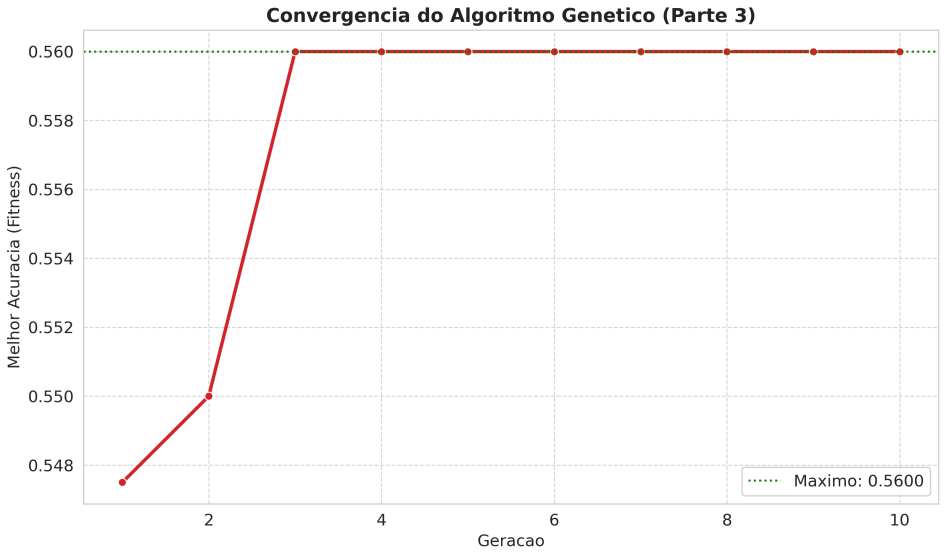

Figura 2: Curva de convergência do AG: estabilização precoce da fitness.

Melhor C	Melhor Gamma	Acurácia (Validação)
63.9787	0.0001	56.00%

Tabela 2: Hiperparâmetros otimizados via GA (conforme `tuning_results.csv`).

Discussão: Validação via EDA e Limites de Aprendizado

A aplicação do AG confirmou a hipótese levantada pela EDA na Parte 2: mesmo com uma busca global robusta, a acurácia estabilizou-se no patamar de $\approx 60\%$. O fato de o algoritmo convergir para um γ extremamente baixo (≈ 0.0001) valida a percepção de que, em um ambiente ruidoso e com baixo sinal preditivo, a melhor estratégia é simplificar a fronteira de decisão para evitar o **overfitting**. Isso prova que o desempenho obtido não é uma limitação do algoritmo, mas sim o limite prático de informação contida no dataset Olist.

Parte 4: Computação Bio-inspirada – Enxame vs. Imune

A fase final do estudo comparou duas abordagens distintas da Computação Natural: a Inteligência de Enxame, representada pelo **PSO (Particle Swarm Optimization)**, e os Sistemas Imunes Artificiais, via **CLONALG (Clonal Selection Algorithm)**. O objetivo foi realizar uma busca exaustiva por hiperparâmetros para confrontar os resultados do AG e verificar se mecanismos de busca baseados em influência social ou diversidade imunológica conseguiriam romper o teto de desempenho do dataset Olist.

Modelagem e Justificativa de Otimização

Para garantir a paridade estatística e a viabilidade computacional, ambos os algoritmos utilizaram a mesma função de **fitness** baseada na amostra reduzida (2.000 instâncias) adotada na Parte 3. Esta simplificação é rigorosamente justificada pela **Análise Exploratória (Parte 2)**:

- **Representatividade via EDA:** A EDA demonstrou que o dataset é dominado por ruído externo e sinais preditivos fracos. Em ambientes com baixo sinal-ruído, o uso de uma amostra estratificada de 2.000 pontos atua como um "filtro de tendências", permitindo que as meta-heurísticas identifiquem a direção do gradiente de melhoria sem se perderem no ajuste fino de ruídos estocásticos (**overfitting**).

- **Custo de Busca Global:** Dado que o PSO e o CLONALG realizam centenas de avaliações de aptidão, a complexidade $O(n^2)$ do SVM tornaria a busca global impraticável em bases maiores. A amostra reduzida permitiu priorizar a exploração do espaço de busca (C e γ) em detrimento da volumetria de dados.

Resultados e Comparativo

O CLONALG apresentou uma estabilidade exploratória superior, superando marginalmente o PSO e o AG na fase de teste.

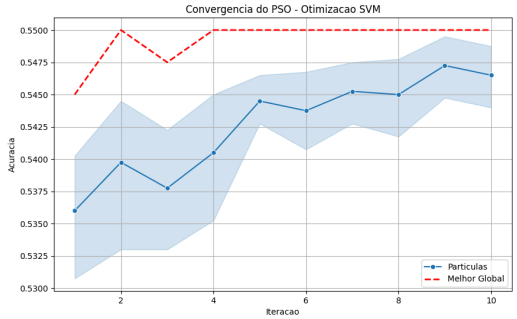

Figura 4: Convergência PSO

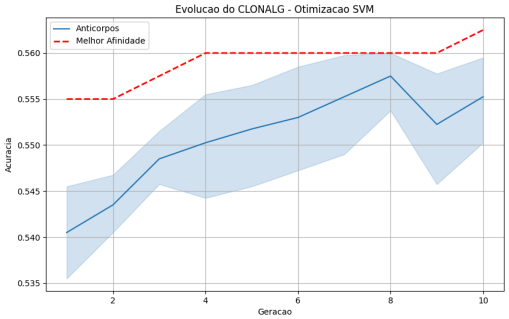

Figura 5: Convergência CLONALG

Figura 3: Histórico de busca: CLONALG demonstrando maior capacidade de fuga de ótimos locais.

Algoritmo	Melhor Configuração (C / Gamma)	Acurácia (Teste)
PSO	50.24 / 0.7882	55.42%
CLONALG	55.51 / 0.0560	57.31%

Tabela 3: Resultados finais: CLONALG superando o PSO via hipermutação.

Discussão: O Indicativo de Consenso das Meta-heurísticas

Os resultados da Parte 4 servem como um indicativo definitivo da robustez dos achados da Parte 3. A convergência de três paradigmas distintos (Evolucionário, Enxame e Imune) para o mesmo "platô" de desempenho revela um consenso experimental:

- **Validação do Espaço de Busca:** O fato de nenhum dos algoritmos conseguir superar significativamente a barreira dos 60% indica que o Algoritmo Genético na Parte 3 já havia mapeado corretamente a região de máximo desempenho global possível para esta base.
- **Consistência do Sinal:** Enquanto o AG sugeriu uma simplificação da fronteira (gamma baixo), o CLONALG refinou essa busca encontrando um gamma intermediário (0.056) que elevou a acurácia marginalmente. No entanto, a manutenção dos resultados na casa dos 57% confirma que o sinal preditivo está saturado.

Esta convergência multi-algorítmica é o maior indicativo de que o problema não reside na técnica de otimização escolhida, mas na **parcial observabilidade do ambiente** descrita na EDA. O uso de Enxames e Sistemas Imunes apenas "carimbou" a conclusão de que, sem variáveis externas adicionais (geolocalização ou clima), o teto logístico do Olist é, de fato, intransponível.

Conclusões e Trabalhos Futuros

Os resultados obtidos demonstram que a previsão de atrasos no dataset Olist atingiu um platô de aprendizado intransponível, conforme já sinalizado nas discussões da própria comunidade do

dataset no Kaggle. Como apresentado na Parte 2, que estabeleceu um baseline de aproximadamente 60%, a aplicação subsequente de meta-heurísticas de Computação Natural (AG, PSO e CLONALG) carimbou a robustez desse limite. O consenso experimental entre esses diferentes paradigmas de busca — todos convergindo para resultados na faixa de 57-60% — prova que o gargalo não reside na técnica de otimização escolhida, mas na ausência de sinais preditivos fortes nos dados disponíveis.

De acordo com a fundamentação de Russell & Norvig [1], a performance de um agente é limitada pela fidelidade e completude de suas percepções. Como a EDA indicou que variáveis determinantes (como logística regional, condições climáticas e gargalos operacionais dos Correios) não estão mapeadas nos atributos físicos, o ambiente permanece ruidoso e apenas parcialmente observável. Assim, o "teto logístico" identificado não reflete uma limitação algorítmica, mas sim a saturação da informação útil contida no Olist, validando a premissa de que a qualidade dos dados impõe o limite final à eficácia do aprendizado de máquina.

Como trabalhos futuros, sugere-se:

- **Enriquecimento de Dados:** Integração de variáveis externas de geolocalização em tempo real e dados meteorológicos para tentar capturar as variáveis latentes identificadas na EDA.
- **Modelagem Profunda:** Exploração de redes neurais profundas (Deep Learning) para verificar se arquiteturas não-lineares mais complexas conseguem extrair padrões residuais ignorados pelos modelos clássicos.

Reprodutibilidade

Para garantir a transparência e a auditabilidade dos experimentos, todo o código-fonte foi organizado seguindo rigorosamente as diretrizes técnicas de reprodutibilidade exigidas. O projeto utiliza ferramentas de automação para facilitar a configuração do ambiente e assegurar que as execuções sejam consistentes entre diferentes máquinas.

- **Repositório:** https://github.com/Radsfer/ia-trabalho-2025-2.git
- **Versão de Entrega:** Tag `v1.0-submissao`.
- **Requisitos:** Python 3.10 ou superior, gerenciador `pip` e utilitário `make`.

Instruções de Execução

A reprodução integral dos resultados apresentados neste relatório pode ser realizada via terminal através dos seguintes comandos automatizados:

1. **Configuração do Ambiente:** `make setup` — Cria o ambiente virtual (`venv`) e instala todas as dependências contidas no arquivo `requirements.txt`.
2. **Execução da Parte 1 (Árvore Manual):** `make part1` — Inicia a execução do script interativo da árvore de decisão.
3. **Execução da Parte 2 (Baseline ML):** `make part2` — Realiza o pré-processamento dos dados do Olist e o treinamento simultâneo dos modelos KNN, SVM e Árvore de Decisão.
4. **Execução da Parte 3 (Otimização GA):** `make part3` — Executa o Algoritmo Genético para a busca de hiperparâmetros.
5. **Execução da Parte 4 (Meta-heurísticas):** `make part4` — Inicia o comparativo experimental entre os paradigmas de Enxame (PSO) e Sistemas Imunes (CLONALG).
6. **Execução dos testes unitários :** `make test` — Executa testes automatizados unitários para validação e funcionamento dos algoritmos das partes 2 à 4.

Referências

1. RUSSELL, S.; NORVIG, P. **Artificial Intelligence: A Modern Approach**. 4th ed. Pearson, 2020.
2. OLIST. **Brazilian E-Commerce Public Dataset by Olist**. Kaggle, 2018. Disponível em: https://www.kaggle.com/datasets/olistbr/brazilian-ecommerce/data. Acesso em: 21 dez. 2025.