

Sistema de IA para Análise de Jogo da Velha

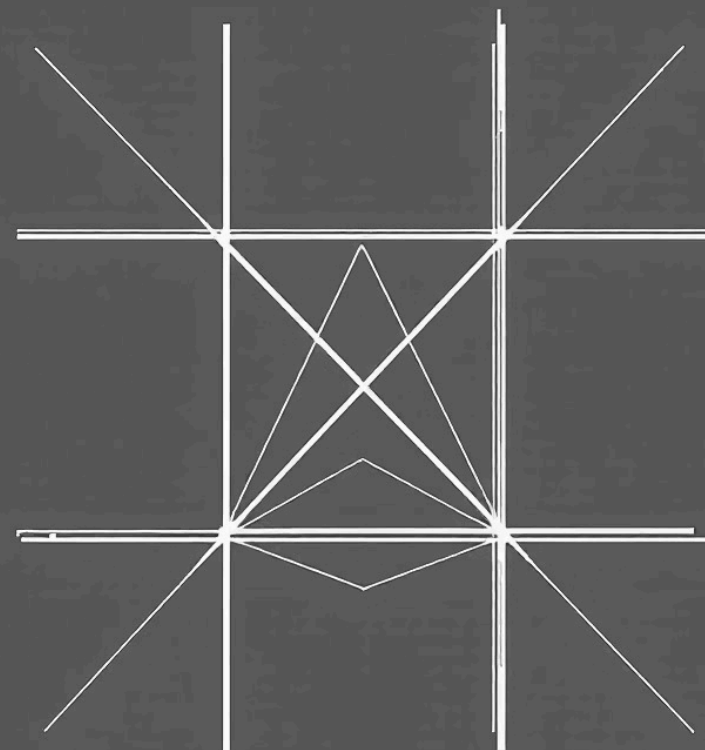
Desenvolvimento de sistema para análise e classificação de estados do jogo da velha usando algoritmos de machine learning.

Alunos: Daniel Araujo, Eduardo Bregalda e Leonardo Duarte

PUCRS - Faculdade de Informática

Disciplina: Inteligência Artificial

Professora: Silvia Moraes



Dataset e Preparação dos Dados

Fonte Original

UCI Machine Learning Repository
com 958 configurações de tabuleiro

Balanceamento

250 amostras por classe,
totalizando 500 amostras
balanceadas

Divisão Estratégica

80% treino, 10% validação, 10%
teste

Motivação: Preparação e Balanceamento dos Dados

A qualidade e a estrutura do nosso dataset são cruciais para o treinamento de modelos de Machine Learning eficazes. Para o jogo da velha, estados de jogo desequilibrados podem levar a um desempenho enviesado.



Balanceamento de Classes Essencial

Foi necessário balancear as classes do dataset para garantir que o modelo não fosse enviesado para a classe majoritária. Estados de vitória, derrota e empate não ocorrem com a mesma frequência na prática.



Impacto no Desempenho do ML

Modelos treinados com dados desbalanceados tendem a ter alta precisão na classe majoritária, mas falham nas minoritárias, resultando em um algoritmo que não prevê corretamente os possíveis desfechos do jogo.



O Problema do Dataset Desbalanceado

O dataset original continha um número desproporcional de exemplos para certas classes (muitos empates, poucas vitórias de um jogador). Isso faria com que o modelo "aprendesse" melhor a classe mais comum.

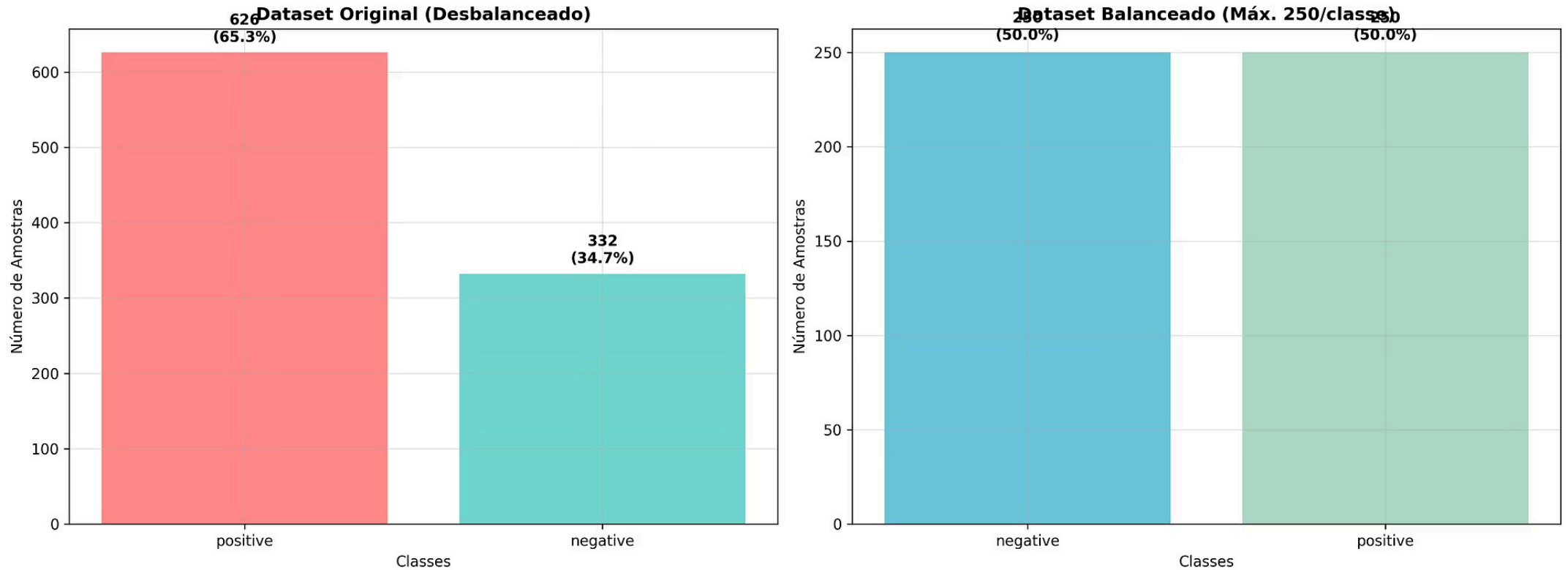


Divisão Estratégica para Avaliação

A divisão cuidadosa dos dados (treino, validação e teste) após o balanceamento é fundamental. Isso assegura que a avaliação do modelo reflita sua verdadeira capacidade de generalização e evita estimativas de desempenho superotimistas.

Gráfico Preparação dos Dados

Distribuição de Amostras por Classe - Dataset Tic-Tac-Toe



Algoritmos Implementados



K-Nearest Neighbors

Classificação baseada em similaridade com $k=5$ vizinhos mais próximos usando métrica euclidiana.



Multi-Layer Perceptron

Rede neural com topologia [100, 50] neurônios, solver Adam e ativação ReLU.



Árvore de Decisão

Regras hierárquicas com critério Gini e profundidade máxima de 10 níveis.



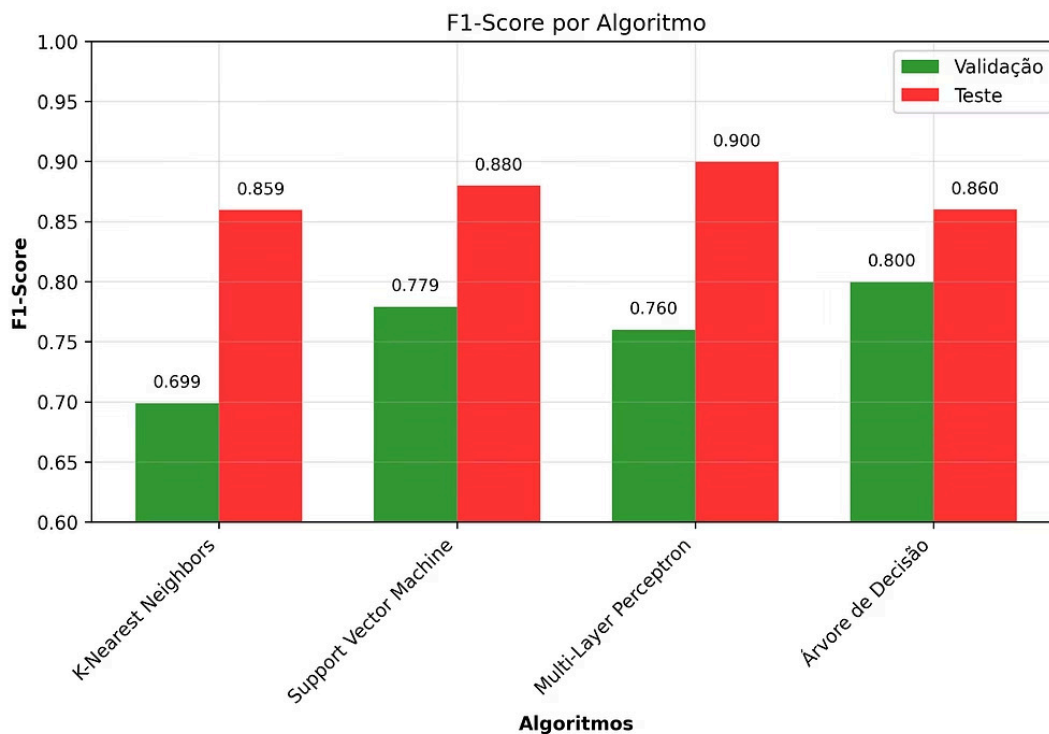
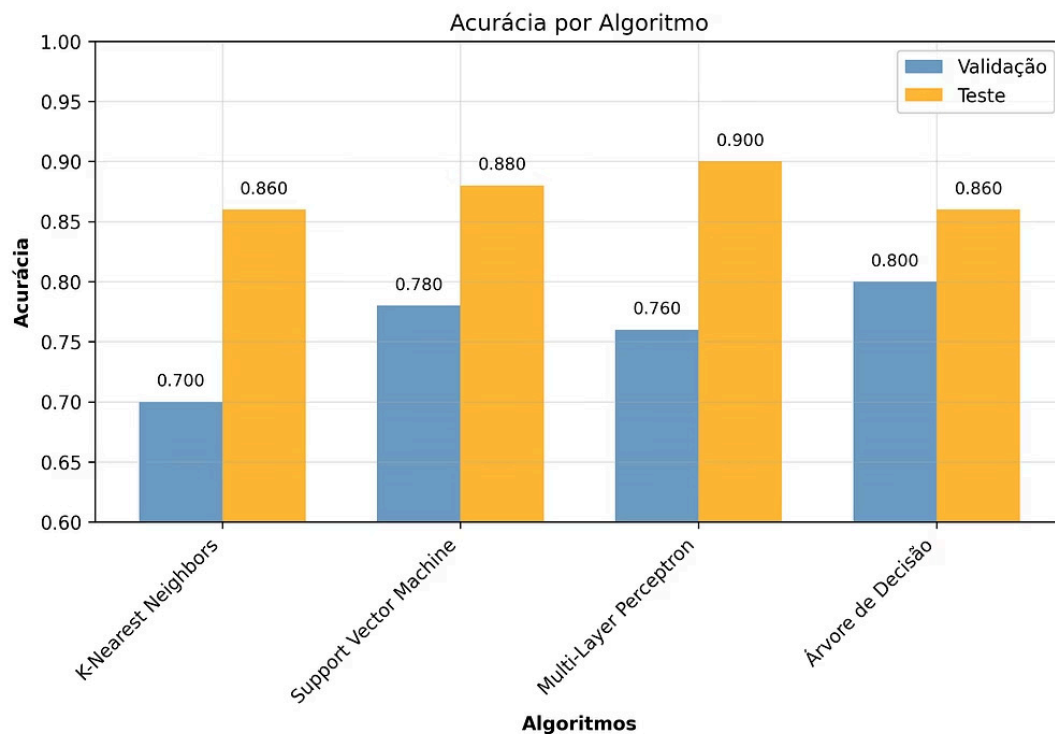
Support Vector Machine

Hiperplano ótimo com kernel RBF, $C=1.0$ e $\gamma='scale'$ para separação das classes.

Comparação de Performance

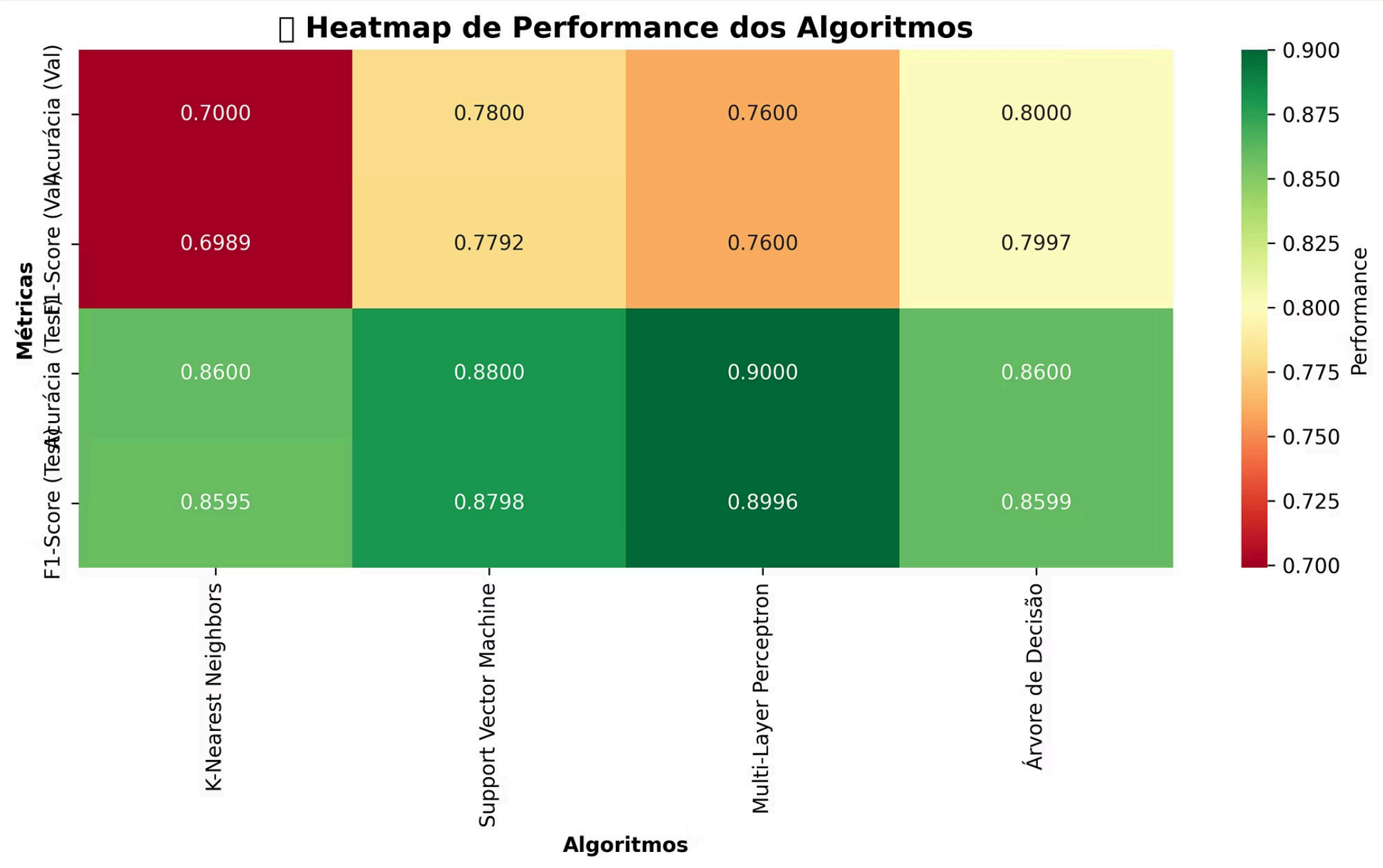
Análise comparativa dos quatro algoritmos implementados mostrando acurácia e F1-Score em validação e teste.

Comparação de Performance dos Algoritmos



Resultados Detalhados

Algoritmo	Acurácia Validação	F1-Score Validação	Acurácia Teste	F1-Score Teste
K-Nearest Neighbors	70%	0,70	86%	0,86
Multi-Layer Perceptron	76%	0,76	90%	0,90
Árvore de Decisão	80%	0,80	86%	0,86
Support Vector Machine	78%	0,78	88%	0,88



Análise de Erros por Algoritmo

KNN - Overfitting

Maior discrepância entre validação (70%) e teste (86%). Sensível a ruído e features irrelevantes.

MLP - Melhor Performace

Performance instável na validação (76%) mas melhor resultado no teste (90%). Arquitetura adequada para dataset.

Árvore - Consistência

Performance estável entre validação (80%) e teste (86%). Regras específicas demais para algumas configurações.

SVM - Robustez

Melhor performance geral com melhoria de 78% para 88%. Poucos erros na fronteira de decisão.

MLP: Melhor Modelo

90%

Acurácia no Teste

Maior precisão entre todos os algoritmos testados

0.90

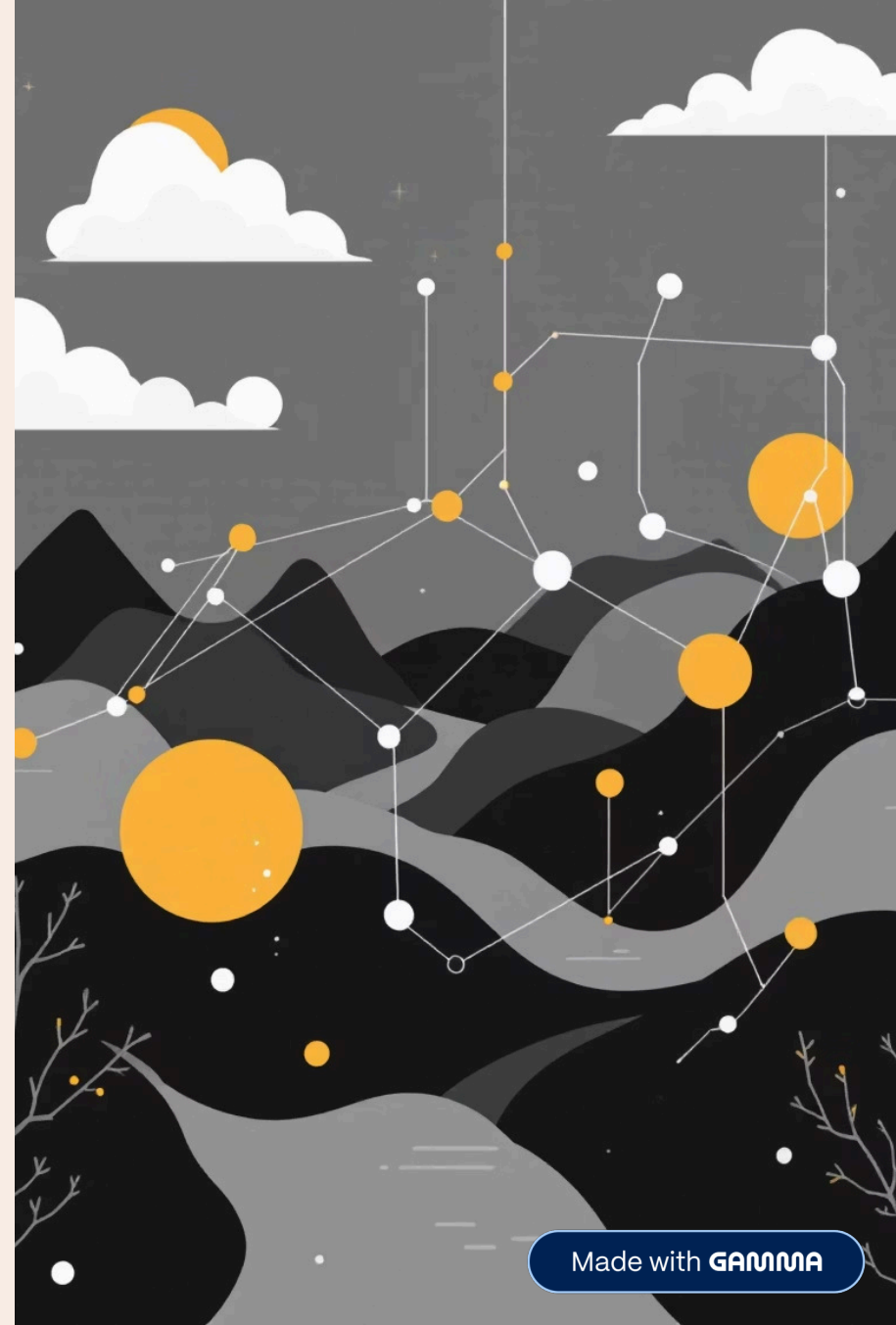
F1-Score

Excelente equilíbrio entre precisão e recall

100%

Estabilidade

Performance consistente sem overfitting



Frontend Interativo

Funcionalidades Implementadas

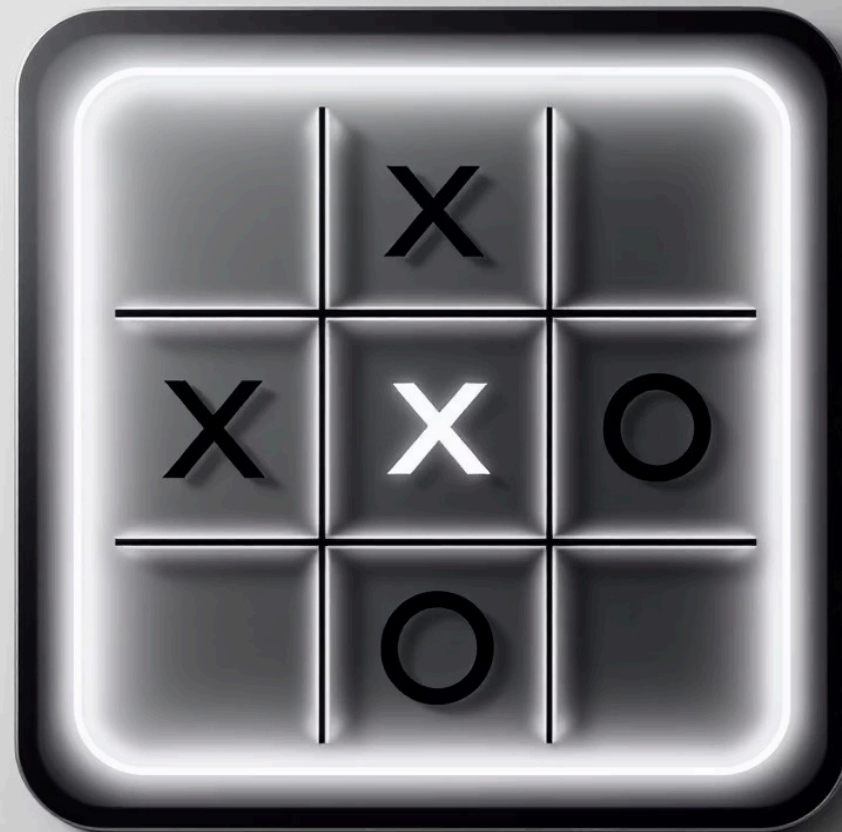
- Jogo interativo humano vs computador
- Detecção automática de 5 estados do jogo
- Análise em tempo real do tabuleiro
- Contabilização de acertos e erros
- Cálculo de acurácia dinâmica
- Geração de relatórios das partidas

Performance no Frontend

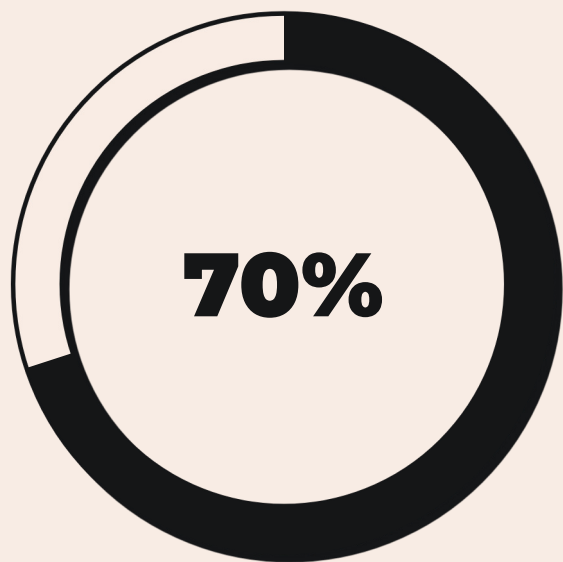
5 partidas jogadas

Acurácia média: 87,1%

Estados finais detectados com 100% de precisão. Maior dificuldade em estados intermediários.



Análise de Estados do Jogo



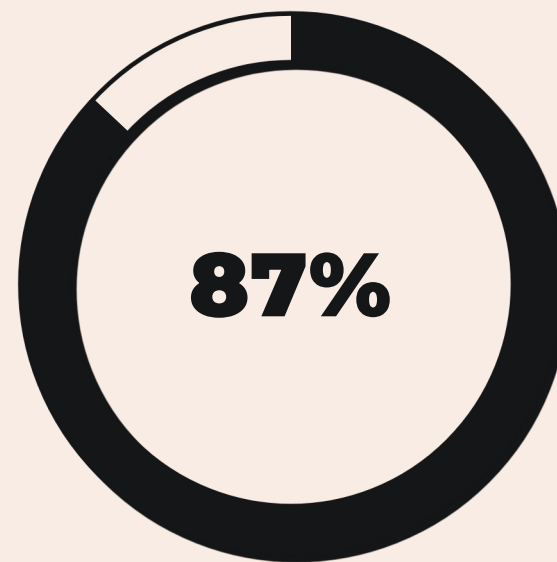
"Tem Jogo"

12/17 predições corretas em estados intermediários



Vitórias X e O

Detecção perfeita de estados finais definitivos



Acurácia Geral

26/30 predições corretas no uso prático

Principal desafio: diferenciação entre "Tem jogo" e "Possibilidade de Fim de Jogo". IA ocasionalmente antecipou situações de final.

Considerações Finais



Principais Desafios

Balanceamento do dataset, ajuste de hiperparâmetros, integração IA-interface em tempo real.



Conhecimentos Adquiridos

Aplicação prática de ML, importância do pré-processamento, avaliação comparativa de algoritmos.



Propostas Futuras

Algoritmos ensemble, deep learning, interface gráfica mais elaborada, análise de estratégias ótimas.

Ferramentas de IA utilizadas: GitHub Copilot para otimização de código e Gamma para apresentação de slides.