

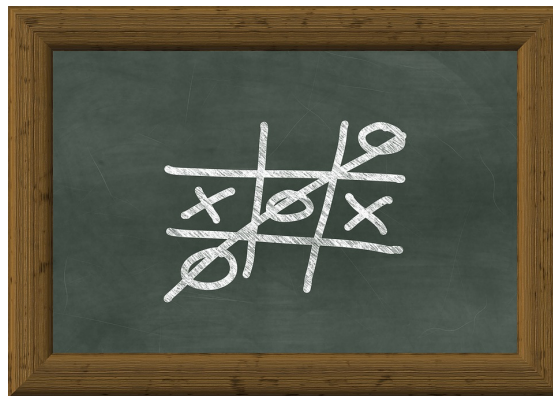
# Instituto Nacional de Telecomunicações - Inatel

## AG002 – Engenharia de Computação

Prof. Me. Marcelo Vinícius Cysneiros Aragão  
Prof. Me. Renzo Mesquita Paranaíba

### 1 Introdução

Neste semestre a AG002 acontecerá na forma de um trabalho prático. Você deverá utilizar seus conhecimentos para, a partir do conjunto de dados proposto, treinar, avaliar e disponibilizar um modelo de aprendizado de máquina para apontar o desfecho de uma partida de jogo da velha.



### 2 Conjunto de Dados

Jogo da velha é um jogo para duas pessoas que requer apenas lápis e papel. O tabuleiro é uma matriz de três linhas por três colunas. Cada jogador se reveza desenhando uma cruz ou um círculo em uma posição desta matriz. O vencedor é aquele que conseguir colocar três peças iguais em uma fileira, na vertical, na horizontal ou na diagonal (conforme ilustrado na figura).

Neste sentido, o conjunto de dados apresenta 958 amostras, que representam todos as possíveis de preencher o tabuleiro do jogo da velha. Cada amostra do conjunto é dada por:

- Nove atributos (enumerados de 1 a 9) que representam o estado de cada posição do tabuleiro; os possíveis valores são  $x$  (cruz),  $o$  (vazio) ou  $o$  (círculo).
- Um rótulo de classe, que representa o desfecho daquela configuração em particular; os possíveis valores são “positivo” (que indica a vitória do  $x$ ) ou “negativo” (que indica empate ou derrota do  $x$ ).

Neste trabalho será utilizada uma versão traduzida do conjunto originalmente concebido por Aha [1] em 1991. Os dados originais foram obtidos do [UCI Machine Learning Repository](#).

### 3 Etapas para Realização

1. Baixar o [conjunto de dados](#) em formato CSV (*comma-separated-values*).
2. Fazer a leitura dos dados utilizando a biblioteca [Pandas](#).
3. Converter os valores presentes no conjunto de dados para números inteiros, de acordo com este mapeamento:  $o \mapsto -1$ ,  $b \mapsto 0$ ,  $x \mapsto +1$ , **negativo**  $\mapsto -1$  e **positivo**  $\mapsto +1$ . Dica: método [replace](#), presente na classe DataFrame do Pandas.
4. Escolher um dos modelos de classificação a seguir:
  - Decision Tree: [Wikipedia](#), [KDnuggets](#) e [scikit-learn](#).
  - k-Nearest Neighbors: [Wikipedia](#), [Towards Data Science](#) e [scikit-learn](#).
  - Multilayer Perceptron: [Wikipedia](#), [KDnuggets](#) e [scikit-learn](#).
  - Naïve Bayes: [Wikipedia](#), [Towards Data Science](#) e [scikit-learn](#).
  - Perceptron: [Wikipedia](#), [Towards Data Science](#) e [scikit-learn](#).
5. [Separar](#) o conjunto de dados em duas partes: 80% para treinamento e 20% para testes.
  - Treinar o modelo escolhido usando 80% dos dados.
  - Avaliar o modelo escolhido usando os 20% restantes.
6. Exibir [métricas de avaliação](#), para que possa ser verificada a acurácia do modelo.
7. Criar uma opção que permita ao usuário inserir dados arbitrários que devem ser classificados pelo modelo. O modelo deverá imprimir se, com base no conhecimento adquirido com os dados do conjunto, os dados inseridos constituem vitória de  $x$  (“sim” ou “não”). Dica: método [predict](#), presente em todos os classificadores.

### 4 Orientações Adicionais

- O trabalho deverá ser feito em dupla;
- Qualquer linguagem de programação pode ser utilizada;
- A entrega deverá ser feita por meio de um arquivo zip com todo o conteúdo do projeto, ou o link de um repositório privado do GitHub;
- Para apresentação, o aluno deverá gravar um vídeo de no máximo 7min de duração, explicando em detalhes as etapas do projeto desenvolvido;
- O vídeo poderá ser feito gravando a própria tela do computador enquanto o aluno explica ou até mesmo ser usado o *smartphone*, desde que as explicações das etapas estejam nítidas;
- A entrega deve ser feita até o dia **26/06/2022**. Disponibilize vídeo e arquivo zip (se for usar) no OneDrive, com permissão de acesso para guilherme@inatel.br. Se usar GitHub (em vez de arquivo zip), disponibilize link também com permissão de acesso.

### Referências

- [1] David W Aha. *Tic-Tac-Toe Endgame database*. Ago. de 1991. URL: <https://archive.ics.uci.edu/ml/datasets/Tic-Tac-Toe+Endgame>.