

Trabalho 2 de inteligência artificial

Integrantes:

Rafael Miranda Oliveira Nascimento;
João Vitor Mota dos Reis;
Henrick Cardoso dos Santos;
Yuri santana oliveira;

1. O Problema: Tomada de Decisão Tática no Futebol

O escanteio é um dos momentos mais cruciais de uma partida, onde ter uma estratégia é muito importante. O problema que o projeto resolve é: **"Dada a configuração atual da defesa e do meu time, qual é a jogada com maior probabilidade de sucesso?"**

Não existe uma resposta única. A decisão depende de múltiplas variáveis (atributos), como:

- A marcação é individual ou por zona?
- O goleiro sai bem do gol?
- Meus jogadores são mais altos que os adversários?
- Existe pressão no cobrador?

2. A Solução: O Algoritmo de Árvore de Decisão (ID3)

O projeto implementa o algoritmo ID3 (Iterative Dichotomiser 3) do zero, construindo o modelo de decisão através da interação contínua entre entropia, ganho de informação e aprendizado recursivo. O sistema primeiro calcula a **entropia** para medir o grau de incerteza das jogadas possíveis com base no histórico de dados. Para minimizar essa incerteza, o algoritmo avalia o **ganho de informação** de cada variável do jogo (como "Pressão" ou "Estilo do Goleiro"), identificando qual atributo separa melhor as situações de sucesso para transformá-lo no nó de decisão daquela etapa. Esse ciclo é movido pelo **aprendizado recursivo** na função `learn_decision_tree`, que repete iterativamente o processo de calcular o ganho, escolher o melhor atributo e ramificar os cenários possíveis, até que a árvore convirja para uma folha final que representa a recomendação tática exata.

3. Correspondência

A seguir está a correspondência do pseudocódigo com o código em python utilizado no projeto.

<pre>function LEARN-DECISION-TREE(examples, attributes, parent_examples) returns a tree if examples is empty then return PLURALITY-VALUE(parent_examples) else if all examples have the same classification then return the classification else if attributes is empty then return PLURALITY-VALUE(examples) else A ← argmax_{a ∈ attributes} IMPORTANCE(a, examples) tree ← a new decision tree with root test A for each value v of A do exs ← {e : e ∈ examples and e.A = v} subtree ← LEARN-DECISION-TREE(exs, attributes - A, examples) add a branch to tree with label (A = v) and subtree subtree return tree</pre>	<pre>def learn_decision_tree(examples, attributes, parent_examples, target_attr="Decision"): if not examples: return plurality_value(parent_examples, target_attr) first_class = examples[0][target_attr] if all(e[target_attr] == first_class for e in examples): return first_class if not attributes: return plurality_value(examples, target_attr) best_attr = max(attributes, key=lambda a: importance(a, examples, target_attr)) tree = {best_attr: {}} possible_values = set(e[best_attr] for e in examples) for v in possible_values: exs = [e for e in examples if e[best_attr] == v] new_attributes = [attr for attr in attributes if attr != best_attr] subtree = learn_decision_tree(exs, new_attributes, examples, target_attr) tree[best_attr][v] = subtree return tree</pre>
--	--

4. Conversas com IA

Abaixo terão duas das interações com IA durante o processo de implementação do código:

Interação 1:

Possuo uma árvore de decisão que auxilia na tomada de decisão de qual seria a melhor jogada de escanteio baseado nas características: 'Marcacao' (Zona ou Mista ou Individual), 'Goleiro' (Fica_Gol ou Sai_Bem), 'Estatura_Nosso_Time' (Baixos ou Altos), 'Estatura_Adversario' (Baixos ou Altos), 'Pressao' (Sim ou Nao). Com isso podendo tomar as decisões de Jogada_Curta, Cruzamento_Ponto_Penalti, Cruzamento_Primeiro_Pau ou Cruzamento_Segundo_Pau. Eu gostaria de aumentar a árvore além de aumentar análise tática de cada jogada, tendo o porque daquela jogada ser a melhor baseado nas característica, as variáveis a serem adicionadas seriam Número de jogadores do nosso time (1-8) e do outro time (1-8) na área, e qualidade do cobrador (1-10).

Resposta 1:

sugeri as duas funções para definir o que seriam os valores das dois novos atributos:

```
def classificar_densidade(nos, eles):
```

```
    diferenca = nos - eles
    if diferenca >= 2:
        return "Superioridade"
    elif diferenca <= -2:
        return "Inferioridade"
    else:
        return "Equilibrado"
```

```
def classificar_cobrador(nota):
```

```
    if nota >= 8:
        return "Elite"
```

```

elif nota >= 5:
    return "Medio"
else:
    return "Fraco"
Atualização dos atributos e o situacao_usuario:
atributos_disponiveis = [
    'Marcacao',
    'Goleiro',
    'Estatura_Nosso_Time',
    'Estatura_Adversario',
    'Pressao',
    'Densidade_Area',
    'Qualidade_Cobrador'
]
'Densidade_Area': classificar_densidade(nossos_jogadores, jogadores_adv),
'Qualidade_Cobrador': classificar_cobrador(nota_cobrador)
Hierarquia final ficou:
Nível 1 – Pressão
Se pressão = Sim
→ prioriza primeiro pau ou jogada curta
Nível 2 – Densidade
Superioridade → cruzar
Inferioridade → evitar duelo
Nível 3 – Estatura
Vantagem aérea → pênalti
Desvantagem → segundo pau ou curto
Nível 4 – Goleiro
Sai bem → evitar bola alta frontal
Fica → explorar área pequena
Nível 5 – Qualidade do cobrador
Elite → segundo pau
Fraco → jogada curta

```

Interação 2:

Crie o dataset atualizado para os novos padrões táticos, lembrando da ordem estratégica ideal da árvore proposta

Resposta 2:

```

dataset_futebol = [

# =====

# ① PRESSÃO = SIM → PRIORIDADE: EXECUÇÃO RÁPIDA

# =====

# Inferioridade sob pressão → evitar duelo

{'Marcacao':'Zona','Goleiro':'Sai_Bem','Estatura_Nosso_Time':'Baixos','Estatura_Adversario':'Altos','Pressao':'Sim','Densidade_Area':'Inferioridade','Qualidade_Cobrador':'Fraco','Decisao':'Jogada_Curta'},

# Equilibrado sob pressão → antecipação

```

```
{'Marcacao':'Individual','Goleiro':'Sai_Bem','Estatura_Nosso_Time':'Altos','Estatura_Adversario':'Altos','Pressao':'Sim','Densidade_Area':'Equilibrado','Qualidade_Cobrador':'Medio','Decisao':'Cruzamento_Primeiro_Pau'},
```

Superioridade sob pressão → atacar rápido

```
{'Marcacao':'Mista','Goleiro':'Fica_Gol','Estatura_Nosso_Time':'Altos','Estatura_Adversario':'Baixos','Pressao':'Sim','Densidade_Area':'Superioridade','Qualidade_Cobrador':'Elite','Decisao':'Cruzamento_Primeiro_Pau'},
```

Goleiro fica e cobrador elite → segundo pau mesmo sob pressão

```
{'Marcacao':'Zona','Goleiro':'Fica_Gol','Estatura_Nosso_Time':'Altos','Estatura_Adversario':'Altos','Pressao':'Sim','Densidade_Area':'Superioridade','Qualidade_Cobrador':'Elite','Decisao':'Cruzamento_Segundo_Pau'},
```

=====

2 PRESSÃO = NÃO → ANÁLISE ESTRUTURAL COMPLETA

=====

A) SUPERIORIDADE NUMÉRICA

Vantagem aérea clara

```
{'Marcacao':'Individual','Goleiro':'Fica_Gol','Estatura_Nosso_Time':'Altos','Estatura_Adversario':'Baixos','Pressao':'Nao','Densidade_Area':'Superioridade','Qualidade_Cobrador':'Medio','Decisao':'Cruzamento_Ponto_Penalti'},
```

Zona + goleiro não sai → explorar segundo pau

```
{'Marcacao':'Zona','Goleiro':'Fica_Gol','Estatura_Nosso_Time':'Altos','Estatura_Adversario':'Altos','Pressao':'Nao','Densidade_Area':'Superioridade','Qualidade_Cobrador':'Elite','Decisao':'Cruzamento_Segundo_Pau'},
```

Goleiro sai bem → bola rápida primeiro pau

```
{'Marcacao':'Mista','Goleiro':'Sai_Bem','Estatura_Nosso_Time':'Altos','Estatura_Adversario':'Altos','Pressao':'Nao','Densidade_Area':'Superioridade','Qualidade_Cobrador':'Medio','Decisao':'Cruzamento_Primeiro_Pau'},
```

Cobrador fraco → reduzir risco

```
{'Marcacao':'Zona','Goleiro':'Sai_Bem','Estatura_Nosso_Time':'Altos','Estatura_Adversario':'Baixos','Pressao':'Nao','Densidade_Area':'Superioridade','Qualidade_Cobrador':'Fraco','Decisao':'Cruzamento_Primeiro_Pau'},
```

B) EQUILIBRADO

Jogo parêlho + vantagem aérea

```
{'Marcacao':'Individual','Goleiro':'Fica_Gol','Estatura_Nosso_Time':'Altos','Estatura_Adversario':'Altos','Pressao':'Nao','Densidade_Area':'Equilibrado','Qualidade_Cobrador':'Medio','Decisao':'Cruzamento_Ponto_Penalti'},
```

Zona congestionada → segundo pau

```
{'Marcacao':'Zona','Goleiro':'Fica_Gol','Estatura_Nosso_Time':'Altos','Estatura_Adversario':'Altos','Pressao':'Nao','Densidade_Area':'Equilibrado','Qualidade_Cobrador':'Elite','Decisao':'Cruzamento_Segundo_Pau'},
```

Desvantagem aérea leve → antecipação

```
{'Marcacao':'Mista','Goleiro':'Sai_Bem','Estatura_Nosso_Time':'Baixos','Estatura_Adversario':'Altos','Pressao':'Nao','Densidade_Area':'Equilibrado','Qualidade_Cobrador':'Medio','Decisao':'Cruzamento_Primeiro_Pau'},
```

Cobrador fraco → jogada construída

```
{'Marcacao':'Individual','Goleiro':'Sai_Bem','Estatura_Nosso_Time':'Baixos','Estatura_Adversario':'Baixos','Pressao':'Nao','Densidade_Area':'Equilibrado','Qualidade_Cobrador':'Fraco','Decisao':'Jogada_Curta'},
```

C) INFERIORIDADE NUMÉRICA

Desvantagem total → evitar duelo

```
{'Marcacao':'Zona','Goleiro':'Sai_Bem','Estatura_Nosso_Time':'Baixos','Estatura_Adversario':'Altos','Pressao':'Nao','Densidade_Area':'Inferioridade','Qualidade_Cobrador':'Fraco','Decisao':'Jogada_Curta'},
```

Inferioridade mas cobrador elite → tentar segundo pau

```
{'Marcacao':'Zona','Goleiro':'Fica_Gol','Estatura_Nosso_Time':'Baixos','Estatura_Adversario':'Altos','Pressao':'Nao','Densidade_Area':'Inferioridade','Qualidade_Cobrador':'Elite','Decisao':'Cruzamento_Segundo_Pau'},
```

Inferioridade leve + marcação individual → primeiro pau

```
{'Marcacao':'Individual','Goleiro':'Sai_Bem','Estatura_Nosso_Time':'Baixos','Estatura_Adversario':'Altos','Pressao':'Nao','Densidade_Area':'Inferioridade','Qualidade_Cobrador':'Medio','Decisao':'Cruzamento_Primeiro_Pau'},
```

Inferioridade mas vantagem aérea isolada

```
{'Marcacao':'Mista','Goleiro':'Fica_Gol','Estatura_Nosso_Time':'Altos','Estatura_Adversario':'Baixos','Pressao':'Nao','Densidade_Area':'Inferioridade','Qualidade_Cobrador':'Medio','Decisao':'Cruzamento_Ponto_Penalti'}
```

]