

08/05/2025
ATIVIDADE AVALIATIVA PRÁTICA
ÁRVORE DE DECISÃO - RESTAURANTE

1. CÓDIGO

```
import pandas as pd
from sklearn.preprocessing import LabelEncoder
from sklearn.tree import DecisionTreeClassifier, export_text
import matplotlib.pyplot as plt
from sklearn import tree

# Dados de exemplo para decisão de restaurante
dados = {
    'orcamento': ['baixo', 'médio', 'médio', 'alto', 'baixo', 'médio', 'alto', 'baixo'],
    'preferenciaDeComida': ['fast food', 'italiano', 'japonês', 'italiano', 'japonês', 'fast food', 'fast food', 'italiano'],
    'distancia': ['próximo', 'próximo', 'longe', 'longe', 'próximo', 'próximo', 'longe', 'longe'],
    'recomendacao': ['sim', 'sim', 'não', 'sim', 'não', 'sim', 'sim', 'não']
}

# Criar DataFrame
df = pd.DataFrame(dados)

# Codificar atributos categóricos
label_encoders = {}
for coluna in df.columns:
    le = LabelEncoder()
    df[coluna] = le.fit_transform(df[coluna])
    label_encoders[coluna] = le

# Separar dados
X = df.drop(columns=['recomendacao'])
y = df['recomendacao']

# Criar modelo e treinar
modelo = DecisionTreeClassifier(criterion='entropy', random_state=42)
modelo.fit(X, y)

# Plotar a árvore de decisão
fig, ax = plt.subplots(figsize=(14, 8))
tree.plot_tree(
    modelo,
    feature_names=X.columns,
    class_names=label_encoders['recomendacao'].classes_,
    filled=True,
    rounded=True,
    proportion=True,
    precision=2,
    fontsize=10,
    ax=ax
)

# Exibir as regras da árvore de decisão
regras = export_text(modelo, feature_names=X.columns.tolist())
print("Regras extraídas da árvore de decisão:\n")
print(regras)

plt.title("Árvore de Decisão - Escolha de Restaurante", fontsize=14, pad=20)
plt.tight_layout()
plt.savefig("arvoreRestaurante.png") # Salva como imagem
plt.show()
```

2. REGRAS

se (orcamento = baixo) e (preferenciaDeComida = fast food) e (distancia=próximo)
então recomendacao= sim
se (orcamento = médio) e (preferenciaDeComida = italiano) e (distancia=próximo)
então recomendacao= sim
se (orcamento = médio) e (preferenciaDeComida = japonês) e (distancia=longe)
então recomendacao= não
se (orcamento = alto) e (preferenciaDeComida = italiano) e (distancia=longe) então
recomendacao= sim
se (orcamento = baixo) e (preferenciaDeComida = japonês) e (distancia=próximo)
então recomendacao= não
se (orcamento = médio) e (preferenciaDeComida = fast food) e (distancia=próximo)
então recomendacao= sim

3. EXPLICAÇÃO

Grupo 3: orçamento = médio, preferência = japonês, distância = longe →
Decisão: Não recomendar
Pela árvore, observa-se que se a comida é japonês e a distância é longe,
mesmo com orçamento médio, não recomendamos, pois não valeria a pena

Segue o link do github com o código, a imagem da árvore de decisão:

https://github.com/kaylannesantos/ifpi/tree/main/top_especiais_computo%C3%A7%C3%A3o-2025.1/atvAvaliativa1