

# PEL202 - Fundamentos de Inteligência Artificial

## Relatório Árvore de Decisão

Matheus R. Teixeira

20 de maio de 2024

### 1. Introdução

O conjunto de dados Iris é um conjunto de dados clássico em aprendizado de máquina. Ele consiste em 150 instâncias de flores de íris, cada uma caracterizada por quatro recursos: a largura e o comprimento de sua sépala e a largura e o comprimento de suas pétalas. O objetivo é classificar cada íris em uma das três classes: Iris-setosa, Iris-versicolor ou Iris-virginica.

### 2. Descrição do Problema

O objetivo deste estudo é implementar um classificador de árvore de decisão para classificar com precisão as espécies de flores de íris com base nos recursos fornecidos. O desempenho do classificador será avaliado usando métricas de avaliação padrão.

### 3. Metodologia

A metodologia envolve os seguintes passos:

1. Carregar e pré-processar o conjunto de dados.
2. Dividir o conjunto de dados em conjuntos de treinamento e teste.
3. Treinar um classificador de árvore de decisão nos dados de treinamento.
4. Avaliar o classificador nos dados de teste usando acurácia, precisão, recall, F1-Score e matriz de confusão.

### 4. Resultados

O classificador de árvore de decisão foi treinado e avaliado no conjunto de dados Iris. O resultado é mostrado abaixo:

## 4.1 Matriz de Confusão

	Setosa	Versicolor	Virginica
Setosa	15	0	0
Versicolor	0	12	3
Virginica	0	0	15

Tabela 1: Matriz de Confusão para o conjunto de dados Iris

## 4.2 Relatório de Classificação

	Precisão	Recall	F1-Score
Setosa	100%	100%	100%
Versicolor	100%	80%	89%
Virginica	83%	100%	91%
Acurácia	93%		

Tabela 2: Relatório de Classificação para o conjunto de dados Iris

## 5. Conclusão

O modelo de classificação baseado na árvore de decisão teve um bom desempenho no conjunto de dados Iris, demonstrando alta precisão na identificação das três espécies de flores de íris. O desenvolvimento adicional poderia incluir a experimentação de diferentes classificadores e técnicas de pré-processamento para melhorar o desempenho.

## 6. Código

```

1 #!/usr/bin/env python
2 # coding: utf-8
3
4 # ## Importa o das Bibliotecas Necessarias
5 #
6
7 # In[1]:
8
9
10 from sklearn.datasets import load_iris
11 from sklearn.model_selection import train_test_split
12 from sklearn.tree import DecisionTreeClassifier

```

```
13 from sklearn.metrics import classification_report, accuracy_score,
    confusion_matrix
14 import pandas as pd
15 import matplotlib.pyplot as plt
16 import seaborn as sns
17
18
19 # ## Carregamento do Dataset Iris
20 #
21
22 # In[2]:
23
24
25 iris = load_iris()
26 X = iris.data
27 y = iris.target
28
29
30 # ## Divis o do Dataset em Treino e Teste
31 #
32
33 # In[3]:
34
35
36 X_train, X_test, y_train, y_test = train_test_split(X, y,
    test_size=0.3, random_state=42, stratify=y)
37
38
39 # ## Cria o e Treinamento do Classificador de rvore de
    Decis o
40 #
41
42 # In[4]:
43
44
45 clf = DecisionTreeClassifier(random_state=42)
46
47 clf.fit(X_train, y_train)
48
49
50 # ## Previs es e Avalia o do Modelo
51 #
52
53 # In[5]:
54
55
56 y_pred = clf.predict(X_test)
57
58 accuracy = accuracy_score(y_test, y_pred)
```

```
59 conf_matrix = confusion_matrix(y_test, y_pred)
60 class_report = classification_report(y_test, y_pred, target_names=
    iris.target_names)
61
62
63 # ## Impressão das Matrizes de Avaliação
64 #
65
66 # In[6]:
67
68
69 print(f"Accuracy: {accuracy:.2f}")
70 print("Confusion Matrix:")
71 print(conf_matrix)
72 print("Classification Report:")
73 print(class_report)
```