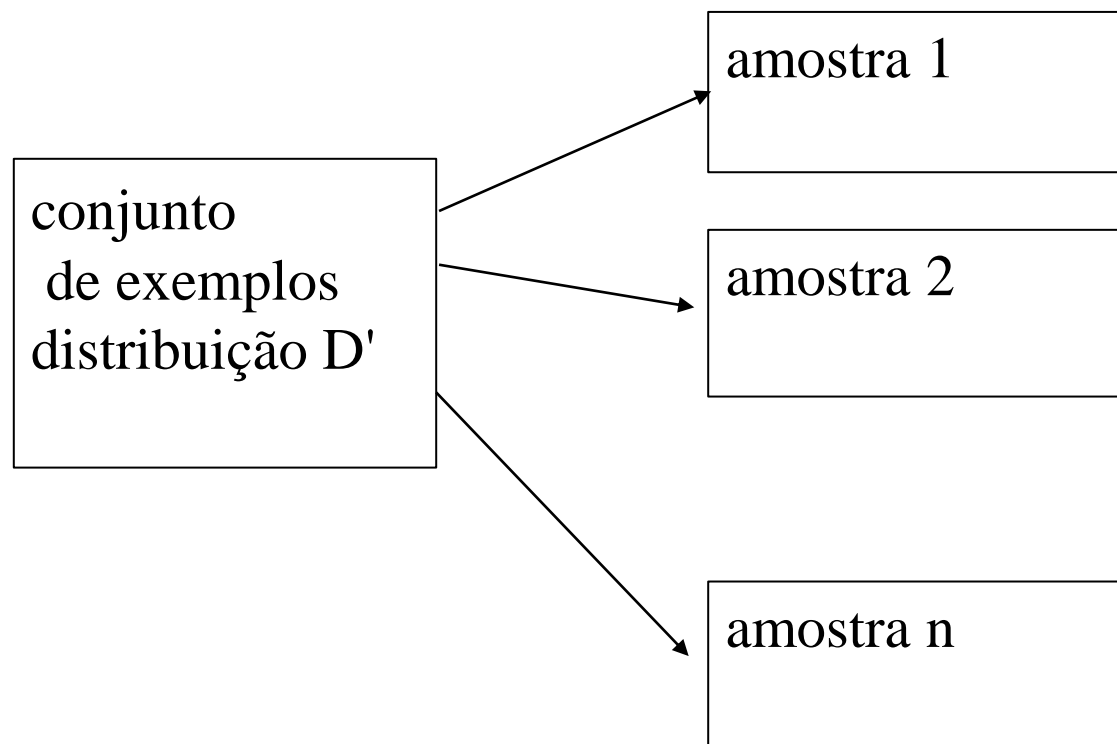


Aprendizado de Máquina
Aprendizado Supervisionado e Não Supervisionado
Avaliação de Classificação

Inteligência Artificial – 2020/1

Avaliação de algoritmos de aprendizado supervisionado

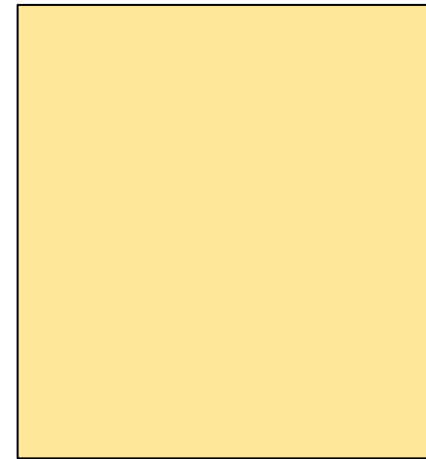
Normalmente baseada na ideia de *amostragem*: seleção de subconjuntos do conjunto de dados inicial, para separar dados de treinamento e dados de teste



Avaliação de algoritmos de aprendizado supervisionado

Métodos de amostragem

- resubstituição:
 - construir o classificador e testar seu desempenho no mesmo conjunto de exemplos
- holdout:
 - divide os exemplos em uma porcentagem fixa de exemplos p para treinamento e $(1-p)$ para teste, considerando normalmente $p > 1/2$



Treinamento e
Teste (mesmo conjunto)



Treinamento

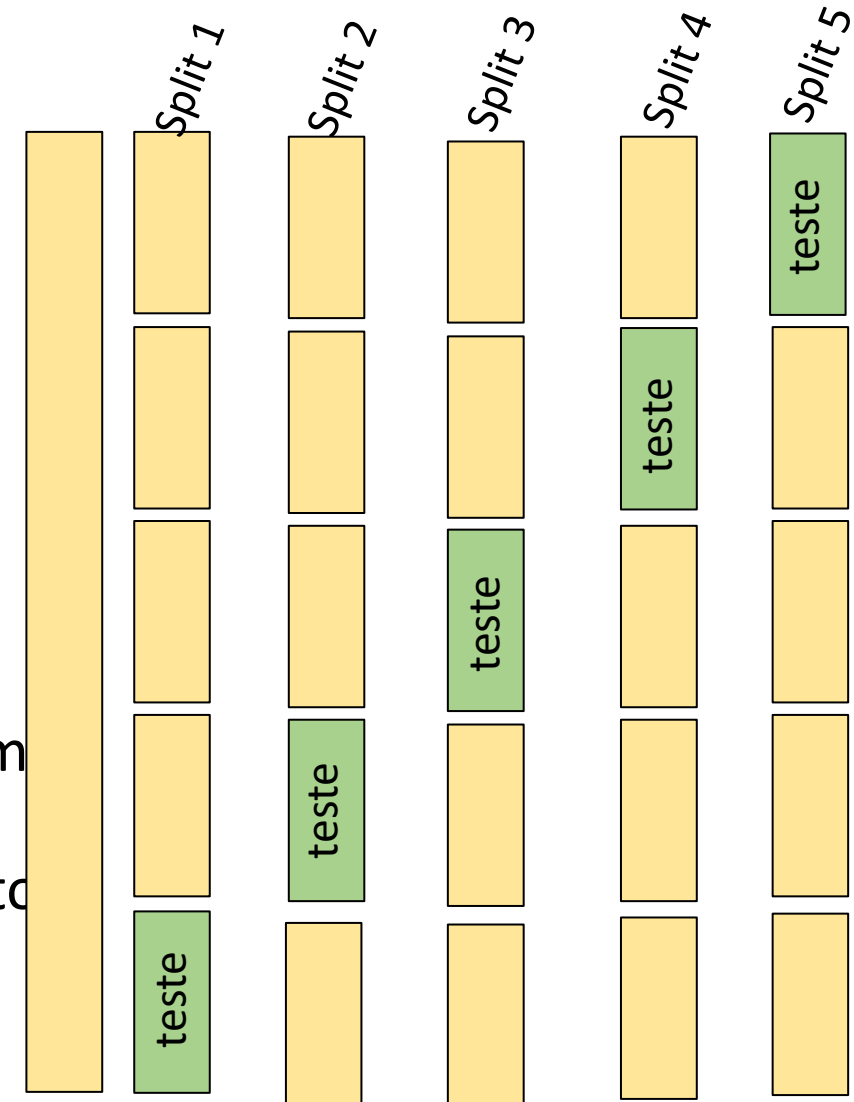


Teste

Avaliação de algoritmos de aprendizado supervisionado

Métodos de amostragem

- Cross-validation (Validação cruzada):
 - r-fold cross validation: exemplos são aleatoriamente divididos em r partições mutuamente exclusivas (folds) de tamanho aproximadamente igual a n/r
 - os exemplos em $r-1$ folds são usados para treinamento e o fold remanescente é usado para teste
 - o treinamento é repetido r vezes, cada vez com um fold como teste
 - o erro é a média dos erros de cada treinamento



Avaliação de algoritmos de classificação

- Conjunto de dados: **clima**
- Atributos:
 - Tempo
 - Temperatura
 - Umidade
 - Vento
- Classes: **sim, não**
- Número total de instâncias:
 - sim (9) não (5)
- Algoritmo: J48 (implementação do Weka do C4.5)
- Método de amostragem: validação cruzada com 10 pastas

Clima				
tempo	temperatura	umidade	vento	jogar
sol	85	85	falso	não
sol	80	90	verdadeiro	não
nublado	83	86	falso	sim
chuva	70	96	falso	sim
chuva	68	80	falso	sim
chuva	65	70	verdadeiro	não
nublado	64	65	verdadeiro	sim
sol	72	95	falso	não
sol	69	70	falso	sim
chuva	75	80	falso	sim
sol	75	70	verdadeiro	sim
nublado	72	90	verdadeiro	sim
nublado	81	75	falso	sim
chuva	71	91	verdadeiro	não

Weka – the workbench for machine learning <https://www.cs.waikato.ac.nz/ml/weka/>

Avaliação de algoritmos de classificação

Matriz de confusão

Uma linha para cada **classe real**

Uma coluna para cada **classe definida pelo modelo**

Para o conjunto de dados **clima** (com todo o conjunto de dados):

Classificada como ->	a	b
a = sim	7	2
b = não	3	2

Avaliação de algoritmos de classificação

Tabela de confusão para classificação binária

Distingue 4 categorias de resultados:

- **Verdadeiro Positivo (VP)** (True positive-TP)
 - Casos em que é retornada a classe *sim* e são da classe *sim*
- **Falso Positivo (FP)** (False positive-FP)
 - Casos em que é retornada a classe *sim* e são da classe *não*
- **Verdadeiro Negativo (VN)** (True negative-TN)
 - Casos em que é retornada a classe *não* e são da classe *não*
- **Falso Negativo (FN)** (False Negative-FN)
 - Casos em que é retornada a classe *não* e são da classe *sim*

VP	FN
FP	VN

Classificada como ->	a	b
a = sim	7	2
b = não	3	2

Avaliação de algoritmos de classificação

Métricas de avaliação

Definidas com base nas 4 categorias de valores

- **Acurácia:**
 - (Com que frequência o classificador está correto?)

$$Acurácia = \frac{VerdadeirosPositivos(VP) + VerdadeirosNegativos(VN)}{Total}$$

- **Precisão:**
 - (Daqueles classificados na classe a, quantos eram realmente da classe a?)

$$Precisão = \frac{VerdadeirosPositivos(VP)}{VerdadeirosPositivos(VP) + FalsosPositivos(FP)}$$

VP	FN
FP	VN

Avaliação de algoritmos de classificação

Métricas de avaliação

- **Revocação (recall):**
 - (Daqueles que são da classe a, quantos foram classificados na classe a?)

VP	FN
FP	VN

$$\text{Revocação} = \frac{\text{VerdadeirosPositivos}(VP)}{\text{VerdadeirosPositivos}(VP) + \text{FalsosNegativos}(FN)}$$

- Precisão e Revocação devem ser equilibradas
- Se o classificador não comete nenhum erro, Precisão e Revocação são iguais, com valor 1

Avaliação de algoritmos de classificação

Métricas de avaliação

- **Medida F (F-measure):**
 - Combina as medidas de precisão e revocação (média harmônica)
 - Quanto maior o valor, melhor o classificador

VP	FN
FP	VN

$$MedidaF = 2 * \frac{precisão * revocação}{precisão + revocação}$$

Métricas de avaliação para o Conjunto de dados clima

Acurácia

$$Acurácia = \frac{VerdadeirosPositivos(VP) + VerdadeirosNegativos(VN)}{Total}$$

$$Acurácia = \frac{7+2}{14} = \frac{9}{14} = 0,6429$$

	a	b
a = sim	7	2
b = não	3	2

Precisão

$$Precisão = \frac{VerdadeirosPositivos(VP)}{VerdadeirosPositivos(VP) + FalsosPositivos(FP)}$$

$$Precisão = \frac{7}{7+3} = \frac{7}{10} = 0,7$$

Revocação

$$Revocação = \frac{VerdadeirosPositivos(VP)}{VerdadeirosPositivos(VP) + FalsosNegativos(FN)}$$

$$Revocação = \frac{7}{7+2} = \frac{7}{9} = 0,7778$$

Métricas de avaliação para o Conjunto de dados clima

Medida F:

$$MedidaF = 2 * \frac{precisão * revocação}{precisão + revocação}$$

	a	b
a = sim	7	2
b = não	3	2

$$MedidaF = 2 * \frac{0,7 * 0,7778}{0,7 + 0,7778} = 2 * \frac{0,54446}{1,4778} = 2 * 0,3684 = 0,7368$$

Avaliação de algoritmos de classificação

Conjuntos de dados com mais de duas classes

Conjunto de dados: **iris**

Atributos:

- Comprimento de sépala

- Largura de sépala

- Comprimento de pétala

- Largura de pétala

Classes: **iris-setosa**, **iris-versicolor**, **iris-virginica**

Número total de instâncias: 150 (50 de cada classe)

Algoritmo: J48 (implementação do Weka do C4.5)

Método de amostragem: holdout 60/40

```
5.1,3.5,1.4,0.2,Iris-setosa
4.9,3.0,1.4,0.2,Iris-setosa
4.7,3.2,1.3,0.2,Iris-setosa
4.6,3.1,1.5,0.2,Iris-setosa
5.0,3.6,1.4,0.2,Iris-setosa
5.4,3.9,1.7,0.4,Iris-setosa
4.6,3.4,1.4,0.3,Iris-setosa
...
7.0,3.2,4.7,1.4,Iris-versicolor
6.4,3.2,4.5,1.5,Iris-versicolor
6.9,3.1,4.9,1.5,Iris-versicolor
5.5,2.3,4.0,1.3,Iris-versicolor
6.5,2.8,4.6,1.5,Iris-versicolor
5.7,2.8,4.5,1.3,Iris-versicolor
6.3,3.3,4.7,1.6,Iris-versicolor
...
6.5,3.0,5.8,2.2,Iris-virginica
7.6,3.0,6.6,2.1,Iris-virginica
4.9,2.5,4.5,1.7,Iris-virginica
7.3,2.9,6.3,1.8,Iris-virginica
6.7,2.5,5.8,1.8,Iris-virginica
7.2,3.6,6.1,2.5,Iris-virginica
```

Avaliação de algoritmos de classificação

Matriz de confusão para conjuntos de dados com mais de duas classes

Para o conjunto de dados **iris**

- Matriz construída para os 60 exemplos de treinamento
- VP e VN formam um único grupo (diagonal principal)
- FP: última linha (soma dos valores da coluna, exceto diagonal)
- FN: última coluna (soma dos valores da linha, exceto diagonal)
- Soma dos valores de FP para todas as classes é **igual** a soma dos valores de FN para todas as classes

		Classe predita			FN
		a	b	c	
Classe real	a = Iris-setosa	17	0	0	0
	b = Iris-versicolor	0	19	2	2
	c = Iris-virginica	0	3	19	3
FP		0	3	2	5

Métricas de avaliação para o Conjunto de dados iris

Matriz de confusão por classes

		Classe predita			FN
		a	b	c	
Classe real	a = Iris-setosa	17	0	0	0
	b = Iris-versicolor	0	19	2	2
	c = Iris-virginica	0	3	19	3
FP		0	3	2	5

As métricas de avaliação são calculadas **por classe**

As métricas gerais são obtidas pela **média** das métricas por classe

Métricas de avaliação para o Conjunto de dados iris

Matriz de confusão por classes

- Cada classe tem uma matriz de confusão com duas classes:
 - A própria classe (positiva)
 - As outras classes com valores somados (negativa)

		Classe predita			FN
		a	b	c	
Classe real	a = Iris-setosa	17	0	0	0
	b = Iris-versicolor	0	19	2	2
	c = Iris-virginica	0	3	19	3
FP		0	3	2	5

a	
(VP) 17	(FN) 0
(FP) 0	(VN) 38

b	
(VP) 19	(FN) 2
(FP) 3	(VN) 36

c	
(VP) 19	(FN) 3
(FP) 2	(VN) 36

Métricas de avaliação para o Conjunto de dados íris

Acurácia

a	
(VP) 17	(FN) 0
(FP) 0	(VN) 38

b	
(VP) 19	(FN) 2
(FP) 3	(VN) 36

c	
(VP) 19	(FN) 3
(FP) 2	(VN) 36

$$Acurácia = \frac{VP + VN}{Total}$$

$$Acurácia_a = \frac{17 + 38}{60} = \frac{55}{60} = 0,9167$$

$$Acurácia_b = \frac{19 + 36}{60} = \frac{55}{60} = 0,9167$$

$$Acurácia_c = \frac{19 + 36}{60} = \frac{55}{60} = 0,9167$$

$$Acurácia_geral = \frac{0,9167 + 0,9167 + 0,9167}{3} = 0,9167$$

Métricas de avaliação para o Conjunto de dados íris

Precisão

a	
(VP) 17	(FN) 0
(FP) 0	(VN) 38

b	
(VP) 19	(FN) 2
(FP) 3	(VN) 36

c	
(VP) 19	(FN) 3
(FP) 2	(VN) 36

$$Precisão_a = \frac{17}{17+0} = 1$$

$$Precisão = \frac{VP}{VP + FP}$$

$$Precisão_b = \frac{19}{19+3} = \frac{19}{22} = 0,86$$

$$Precisão_c = \frac{19}{19+2} = \frac{19}{21} = 0,83$$

$$Precisão_geral = \frac{1 + 0,86 + 0,83}{3} = \frac{2,69}{3} = 0,8967$$

Métricas de avaliação para o Conjunto de dados íris

Revocação

a	
(VP) 17	(FN) 0
(FP) 0	(VN) 38

b	
(VP) 19	(FN) 2
(FP) 3	(VN) 36

c	
(VP) 19	(FN) 3
(FP) 2	(VN) 36

$$Revocação = \frac{VP}{VP + FN}$$

$$Revocação_{-a} = \frac{17}{17 + 0} = 1$$

$$Revocação_{-b} = \frac{19}{19 + 2} = \frac{19}{21} = 0,83$$

$$Revocação_{-c} = \frac{19}{19 + 3} = \frac{19}{22} = 0,86$$

$$Revocação_{-geral} = \frac{1 + 0,83 + 0,86}{3} = \frac{2,69}{3} = 0,8967$$

Métricas de avaliação para o Conjunto de dados íris

F-measure

a	
(VP) 17	(FN) 0
(FP) 0	(VN) 38

b	
(VP) 19	(FN) 2
(FP) 3	(VN) 36

c	
(VP) 19	(FN) 3
(FP) 2	(VN) 36

$$MedidaF = 2 * \frac{precisão * revocação}{precisão + revocação}$$

$$MedidaF_{-a} = 2 * \frac{1 * 1}{1 + 1} = 2 * \frac{1}{2} = 1$$

$$MedidaF_{-b} = 2 * \frac{0,86 * 0,83}{0,86 + 0,83} = 2 * \frac{0,7138}{1,69} = 2 * 0,4224 = 0,8447$$

$$MedidaF_{-c} = 2 * \frac{0,83 * 0,86}{0,83 + 0,86} = 2 * \frac{0,7138}{1,69} = 2 * 0,4224 = 0,8447$$

$$MedidaF_{-geral} = \frac{1 + 0,8447 + 0,8447}{3} = \frac{2,6894}{3} = 0,8965$$

Execução e avaliação de classificação em Python

Árvore de decisão em Python

- O algoritmo indutor de árvore de decisão disponível na biblioteca scikit-learn é o algoritmo CART
- **(Não aceita atributos nominais!!)**
- classe `DecisionTreeClassifier` do módulo `sklearn.tree`
`DecisionTreeClassifier(...)`
- Parâmetros de interesse:
- `criterion`
 - Define a função usada para medir a qualidade de uma divisão.
 - Valores possíveis:
 - “gini” – medida de impureza Gini (valor default)
 - “entropy” – ganho de informação.
- `min_samples_split`
 - Número mínimo de amostras necessárias para dividir um nó interno da árvore
 - tipo int or float, default=2



<https://scikit-learn.org>

Árvore de decisão em Python

- Carregar o conjunto de dados Iris, que faz parte do módulo sklearn.datasets

```
from sklearn.datasets import load_iris
```

```
X, y = load_iris(return_X_y=True)
```

- Dividir o conjunto de dados, em treinamento e teste

```
from sklearn.model_selection import train_test_split
```

```
X_train, X_test, y_train, y_test = train_test_split(X, y, random_state=0)
```

- **test_size**: tamanho ou proporção do conjunto de teste
- **train_size**: tamanho ou proporção do conjunto de treinamento

Árvore de decisão em Python

- Instanciar um objeto `DecisionTreeClassifier` com o critério de ganho de informação para a divisão de atributos (entropy)

```
from sklearn import tree
clf = tree.DecisionTreeClassifier(criterion='entropy')
```

- Executar o método **fit()** para gerar o modelo com os parâmetros X (matriz de atributos) e y (classes conhecidas)

```
clf = clf.fit(X_train, y_train)
```

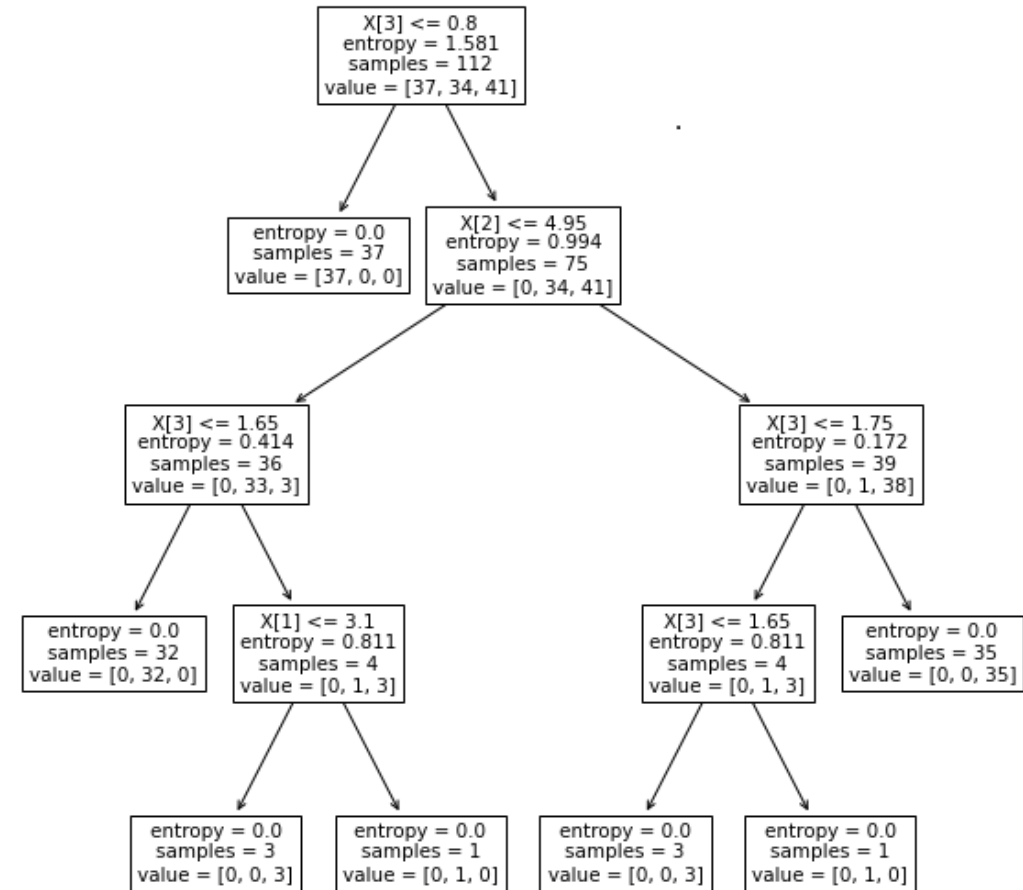

Árvore de decisão em Python

- Depois de gerar o modelo, mostrar a figura da árvore

```
from matplotlib import pyplot as plt
fig = plt.figure(figsize=(10,10))
fig = tree.plot_tree(clf)
```

- Fazer a classificação dos dados de teste

```
predicted = clf.predict(X_test)
```



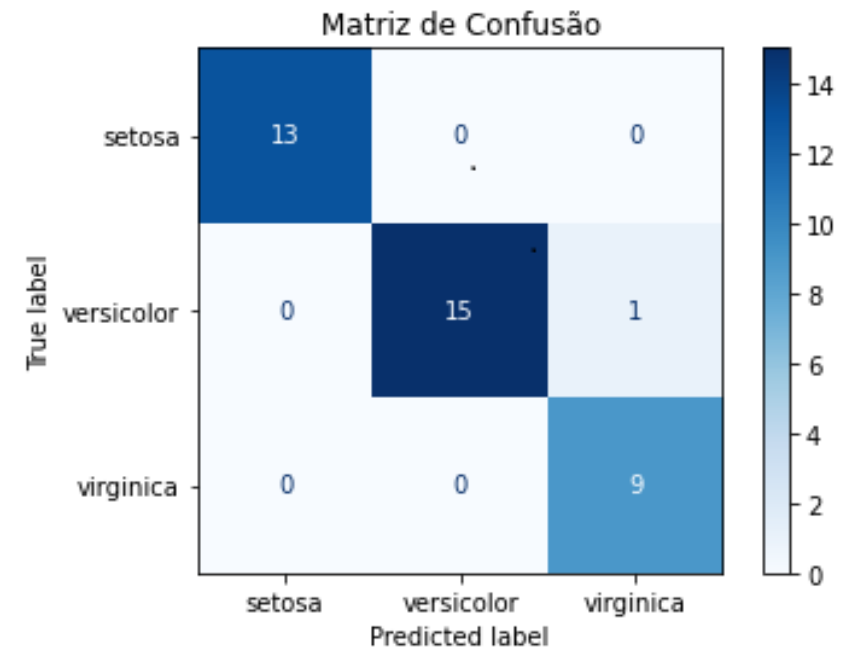
Árvore de decisão em Python

- Encontrar a matriz de confusão

```
from sklearn.metrics import plot_confusion_matrix
class_names = iris.target_names
title = "Matriz de Confusão"
disp = plot_confusion_matrix(clf, X_test, y_test,
                             display_labels=class_names,
                             cmap=plt.cm.Blues)

disp.ax_.set_title(title)

plt.show()
```



Árvore de decisão em Python

- Calcular as medidas de precisão, revocação, medida F e acurácia

```
from sklearn.metrics import classification_report
print("Relatório de avaliação do classificador: \n")
print(f"{classification_report(y_test, predicted)}\n")
```

Relatório de avaliação do classificador:

	precision	recall	f1-score	support
0	1.00	1.00	1.00	13
1	1.00	0.94	0.97	16
2	0.90	1.00	0.95	9
accuracy			0.97	38
macro avg	0.97	0.98	0.97	38
weighted avg	0.98	0.97	0.97	38

FIM