

Árvores de Decisão

Docente:
Prof. Dr. Paulo Fazendeiro

Discentes:

Joana Branco M11020
João Branco M11019
Pedro Moreira M11052

Considerações iniciais

Árvores de Decisão em geral..

01

Tipos de Árvores de Decisão

Árvores de Decisão na AA.

02

Índice

03

Métricas

Algoritmos de cálculo do custo.

04

Usos

Vantagens, Desvantagens e
aplicações.



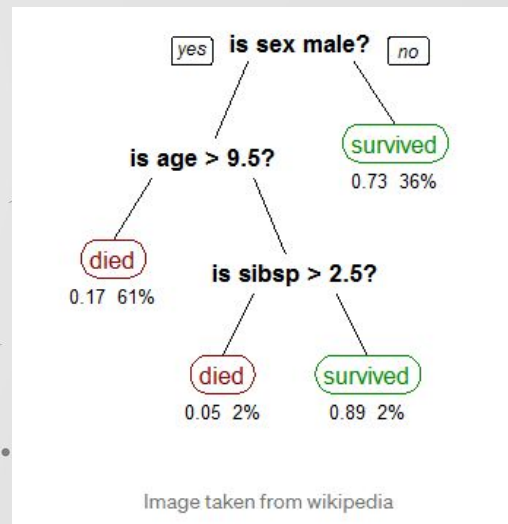
01

Considerações iniciais

Árvores de Decisão no geral

Considerações Iniciais

- Metodologia de **aprendizagem supervisionada**;
- Tem como objetivo prever um valor de uma certa variável através da **aprendizagem de regras de decisão** presentes no *dataset*;
- Usada para **representar visualmente e explicitamente** as decisões e a **tomada de decisão**;
- Quanto **mais profunda** for a árvore, **mais complexas** serão as suas regras de decisão, e **melhor treinado** será o modelo.



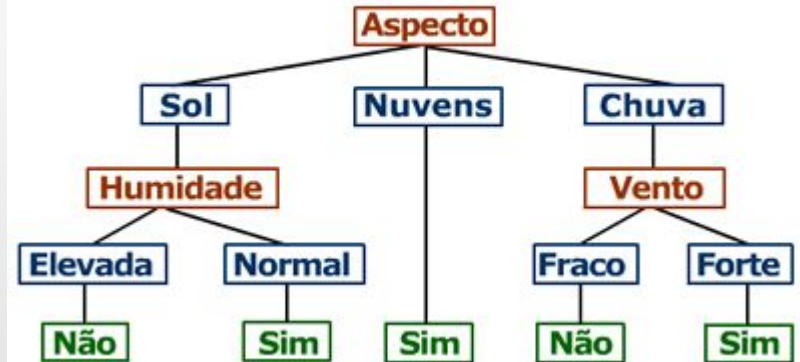
Considerações Iniciais (cont.)

Representação de uma tabela sobre a forma de uma árvore.

Exemplos de Treino

Dia	Aspecto	Temp.	Humidade	Vento	Jogar Ténis
D1	Sol	Quente	Elevada	Fraco	Não
D2	Sol	Quente	Elevada	Forte	Não
D3	Nuvens	Quente	Elevada	Fraco	Sim
D4	Chuva	Ameno	Elevada	Fraco	Sim
D5	Chuva	Fresco	Normal	Fraco	Sim
D6	Chuva	Fresco	Normal	Forte	Não
D7	Nuvens	Fresco	Normal	Fraco	Sim
D8	Sol	Ameno	Elevada	Fraco	Não
D9	Sol	Fresco	Normal	Fraco	Sim
D10	Chuva	Ameno	Normal	Forte	Sim
D11	Sol	Ameno	Normal	Forte	Sim
D12	Nuvens	Ameno	Elevada	Forte	Sim
D13	Nuvens	Quente	Normal	Fraco	Sim
D14	Chuva	Ameno	Elevada	Forte	Não

Árvore de Decisão para Jogar Ténis





02

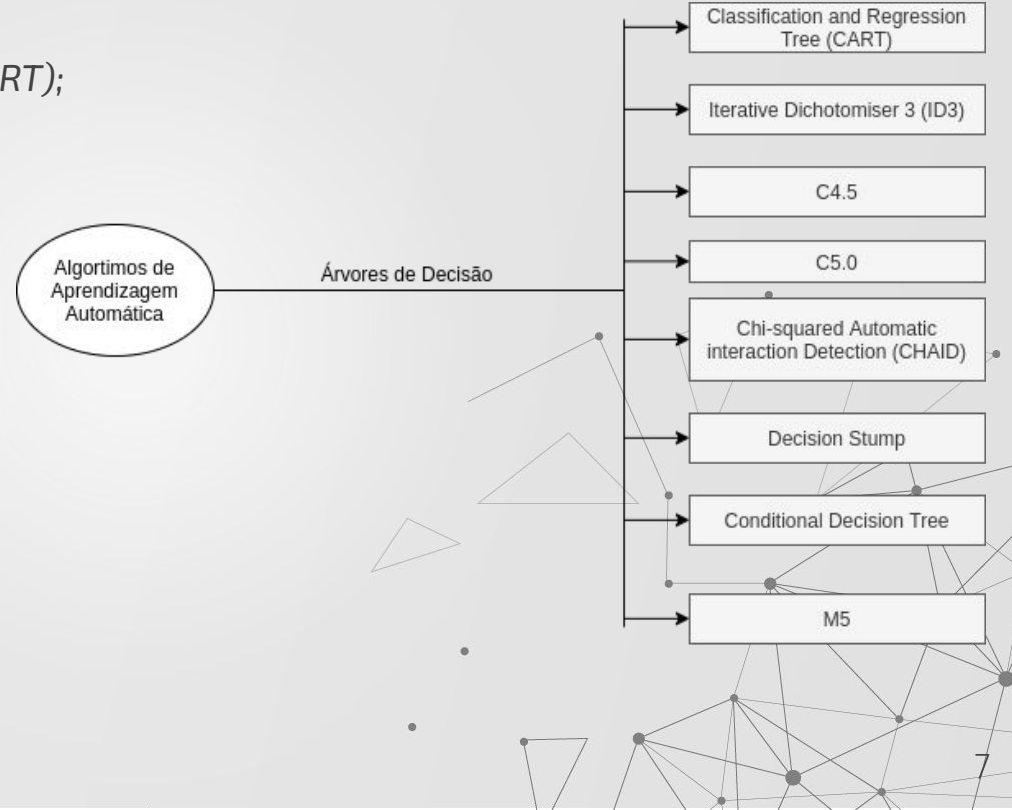
Tipos de Árvores de Decisão

Árvores de Decisão na AA

Algoritmos

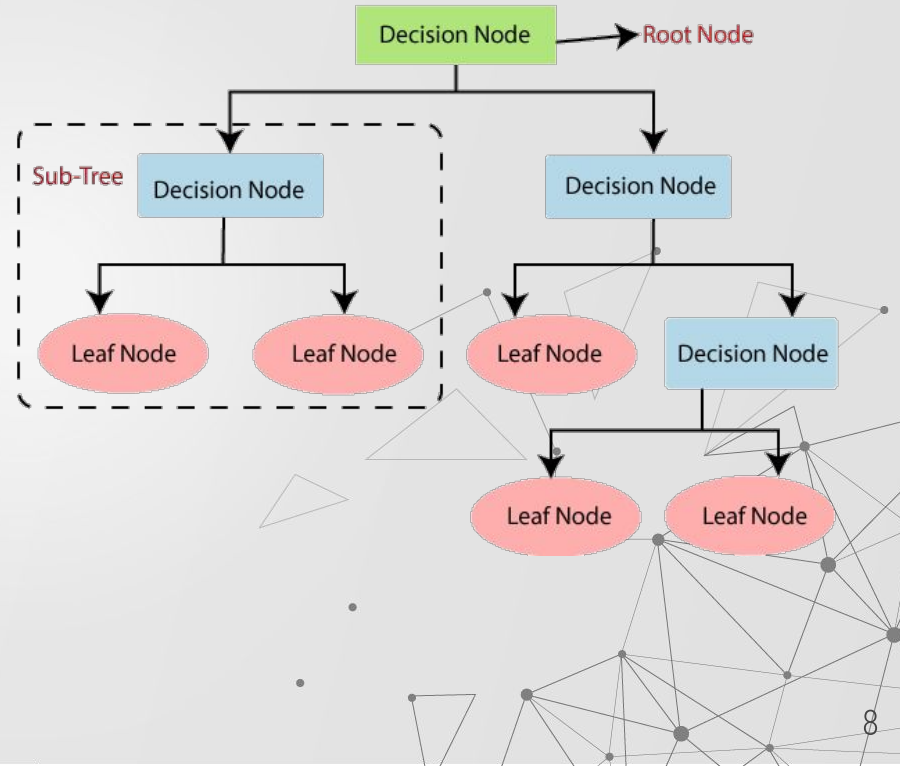
Existem diversos algoritmos, estes são:

- *Classification and Regression Tree (CART);*
- *Iterative Dichotomiser 3 (ID3);*
- *C4.5;*
- *C5.0;*
- *Chi-squared Automatic interaction Detection (CHAID);*
- *Decision Stump;*
- *Conditional Decision Tree;*
- *M5.*



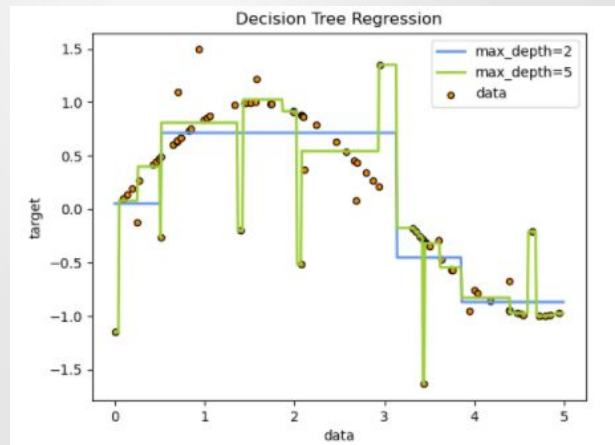
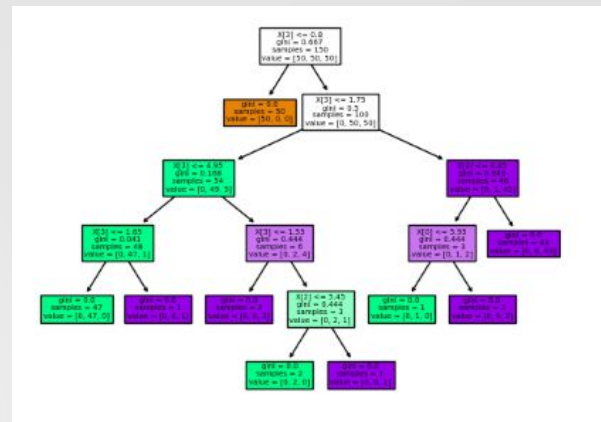
Classificação e Regressão (CART)

- O algoritmo *CART* **começa na raiz da árvore** e **cria dois nodos** no próximo nível da árvore;
- De seguida, o **mesmo procedimento repete-se** para os dois nodos que foram criados, e por aí adiante;
- Este cria uma **árvore alta**, sendo que irá **podar** alguns dos seus **ramos no final do processo**.



Classificação e Regressão (cont.)

	Regressão	Classificação
Output	É um número real.	É uma classe discreta à qual a informação do <i>dataset</i> pertence.





03

Métricas

Cálculo do custo

Gini

- Utilizada em problemas de **Classificação**;
- A função de índices Gini dá-nos uma ideia de quão **bem a separação está feita**, pois mede as vezes em que **um elemento**, escolhido ao acaso, **aparece mal classificado**.

$$G = \sum(pk * (1 - pk))$$

- Uma **classe perfeita** será aquela onde **$pk = 1$** ou **$pk = 0$** e **$G = 0$** , onde os nodos terão uma divisão de **classes 50-50**.



Ganho de Informação

- Utilizada em problemas de **Regressão**;
- Irá ser **calculada para todos os pontos**, onde o **custo será calculado** para todos os **possíveis candidatos a serem separados**;

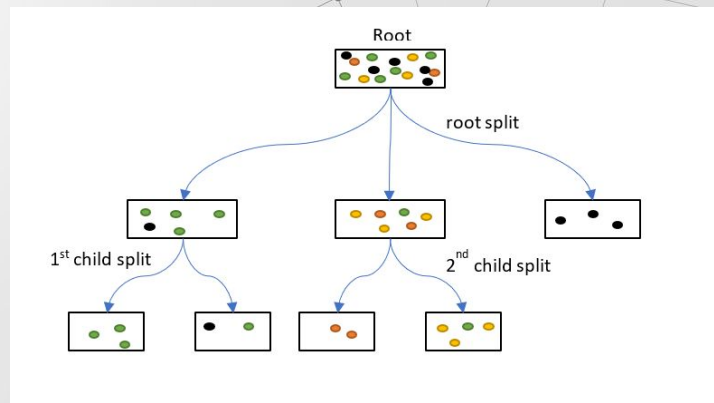
$$\text{sum}(y - \text{prediction})^2$$

- O candidato com **menor custo** será **o escolhido**.



Quando devemos parar a separação?

- Por norma, quanto **maior for o número de atributos, maior será o número de nodos** da árvore. Tais árvores **serão complexas** e podem levar a **overfitting**;
- As formas mais comuns de **combater o overfitting** são:
 - Definição de um **valor mínimo de inputs de treino em cada folha**;
 - Definição de uma **profundidade máxima da árvore**.



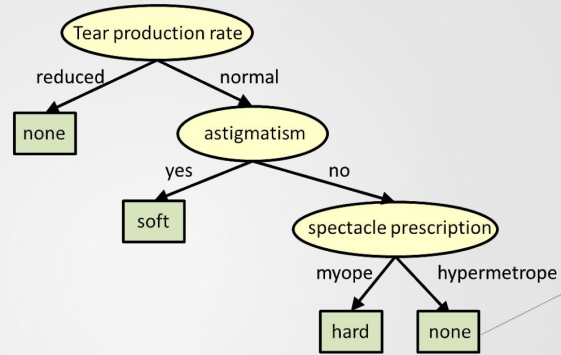
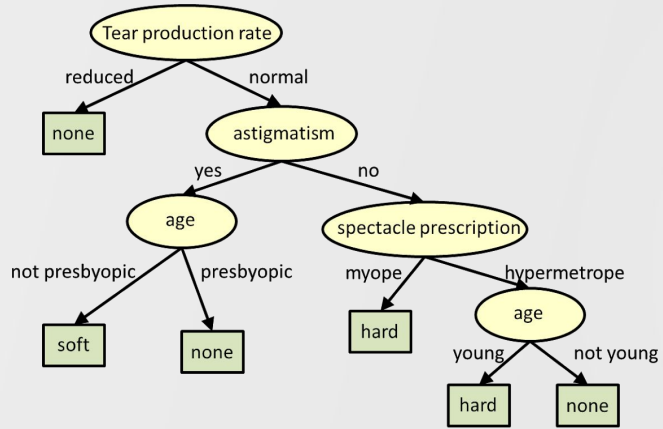
Poda



- A *performance* da árvore pode ser incrementada através da poda, onde para tal **serão removidas as ligações** que fazem **uso de atributos pouco relevantes**.
- Os métodos mais comuns são:
 - Começar nas folhas e ir **removendo os nodos mais populares** de uma classe naquela folha, e verificar se **a mudança não afetou a precisão da mesma** - ***Reduced Error Pruning***;
 - Recorrendo a um parâmetro de aprendizagem, **verificamos o custo de cada nodo**, onde **serão removidos** aqueles com **base no tamanho das suas sub-árvores** - ***Weakest Link Pruning***.



Poda (cont.)





04

Usos

Vantagens, Desvantagens e Aplicações

Vantagens

Compreensão simples

Existe uma simplicidade de compreensão e interpretação

Pouco pré-processamento de dados

Face a outros algoritmos de Classificação e Regressão

Baixo custo

O custo de uso de uma árvore de decisão é uma função logarítmica face aos número de dados.

Modelo *White-box*

Face a outros algoritmos, que recorrem a modelos *Black-box*

Multi-output

Capacidade de lidar com problemas de multiplos *output*.

Não há restrição no tipo de dados

Pode manipular variáveis numéricas e categóricas.

Desvantagens

Difícil Implementação em alguns casos

Certos problemas são difíceis de implementar através de uma árvore de decisão

Overfitting

Pode criar árvores demasiado complexas que não generalizam bem a informação

Maior uso de memória e tempo

O cálculo matemático da árvore de decisão geralmente requer mais memória e tempo

Criação de Árvores desequilibradas

Quando existe uma classe predominante no *dataset*

Treino mais demorado

O tempo de treino do modelo é relativamente maior, devido à sua elevada complexidade

Não adequado para variáveis contínuas

Ao trabalhar com variáveis numéricas contínuas, a árvore de decisão perde informações quando categoriza variáveis em diferentes categorias.

Usos no Mundo Real



Guilherme Nami, Senior Consultant at Simon-Kucher and Partners

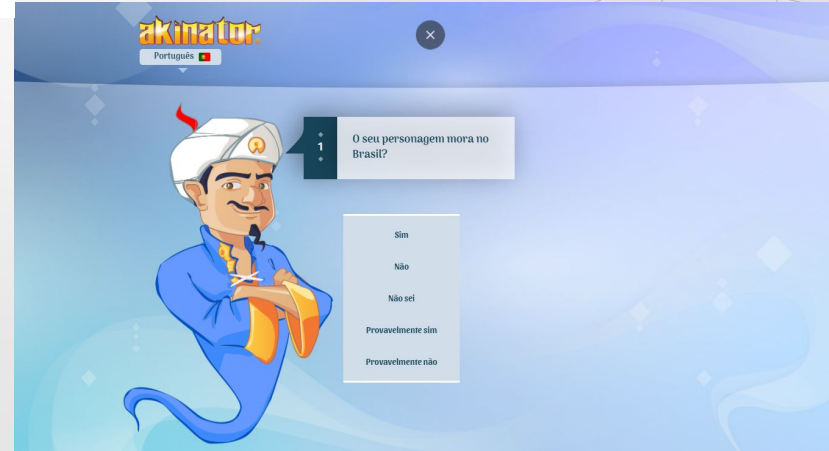
Answered April 15, 2019 · Author has 68 answers and 46.3K answer views

I have personally used them during strategic consulting projects for:

1. Predicting high occupancy dates for hotels
2. Identifying factors leading to better gross margins on a retail chain (curious: # of drugstores nearby was particularly effective for this client)
3. Identifying correlates to high average checks for a global quick-service restaurant chain

The possibilities are, honestly, too many to list.

5.8K views · View 2 Upvoters



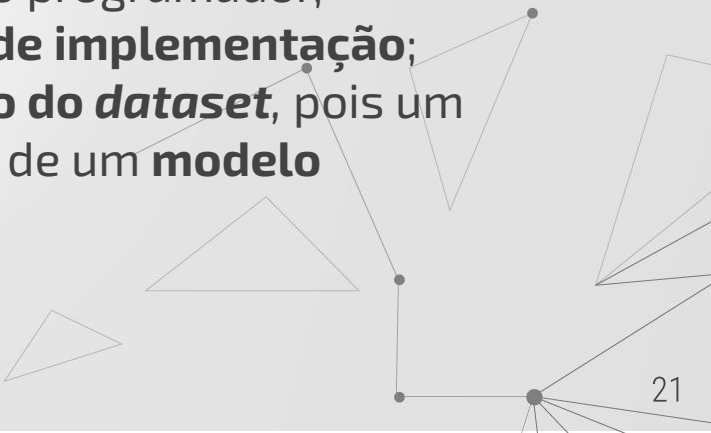
Usos na Prática

```
1 from sklearn import tree
2
3 X = [[0, 0], [1, 1]]
4 Y = [0,1]
5 clf = tree.DecisionTreeClassifier()
6 clf = clf.fit(X, Y)
7
8 out = clf.predict_proba([[2,2]])
9 print(out)
```

```
1 import pandas as pd
2 import os
3 from sklearn import tree
4
5 x_values = []
6 y_values = []
7
8 def readCSV(file_name: str):
9     global x_values, y_values
10    file_path = os.path.join(os.getcwd(), file_name)
11    base = pd.read_csv(file_path)
12
13    x_values = base.iloc[:,0].values
14    x_values = x_values.reshape(-1,1)
15
16    y_values = base.iloc[:,1].values
17    y_values = y_values.reshape(-1,1)
18
19 def Regression():
20     global x_values, y_values
21     readCSV('pizza.csv')
22
23     regressor = tree.DecisionTreeRegressor()
24     regressor.fit(x_values, y_values)
25
26     out = regressor.predict([[100]])
27     print(out)
28
29 Regression()
```



Em resumo...

- A utilização de uma Árvore de Decisão oferece uma **maneira simples** para o acesso ao conhecimento, sendo esta muito importante na **tomada de decisões**;
 - Existem vários algoritmos, sendo que o mais utilizado é o *CART*;
 - É um método de fácil interpretação por parte do programador, devido ao seu **modelo *White-box*** e **facilidade de implementação**;
 - Existe a necessidade de tratar bem a **proporção do *dataset***, pois um ***dataset* desequilibrado** poderá levar à criação de um **modelo também desequilibrado**.
- 



Obrigado.

Questões?