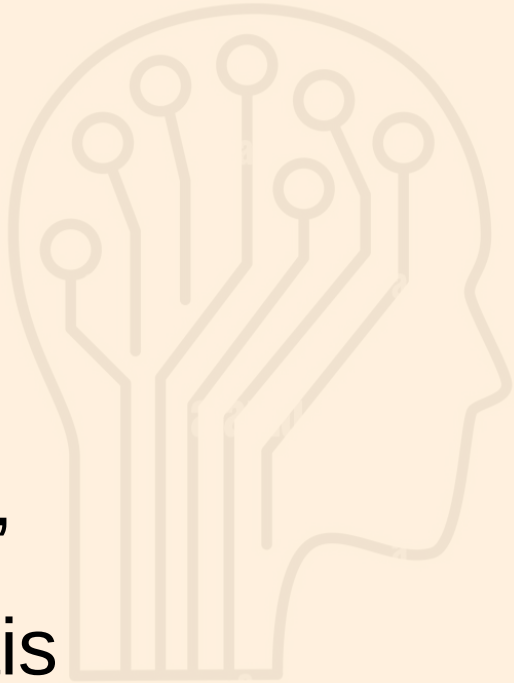
Árvores de decisão



- Aprendizado supervisionado
- Geração de regras a partir de dados
  - Nominais ou numéricos
- Usa a teoria do “ganho de informação”
- A raiz da árvore contém o atributo “mais informativo” ou “mais discriminativo”
  - E vai descendente por ordem deste ganho de informação

Ideia central

- Classificação particionando o espaço de exemplos
- Aprendizado das classes discretas (não contínuas)
  - Valor de **Y** deve ser discreto ou nominal, não deve ser numérico-contínuo

- Um exemplo (clássico da área)
  - Criado pelo idealizador do algoritmo, Ross Quinlan (década de 80)
  - Observar as variáveis climáticas que podem favorecer um jogo de tênis em quadra aberta
  - 14 observações

Jogar tênis

Dia	*Aparência*	*Temperatura*	*Umidade*	*Vento*	*Jogar tênis?*
D1	sol	quente	alta	fraca	**não**
D2	sol	quente	alta	forte	**não**
D3	nublado	quente	alta	fraca	**sim**
D4	chuva	amena	alta	fraca	**sim**
D5	chuva	fria	normal	fraca	**sim**
D6	chuva	fria	normal	forte	**não**
D7	nublado	fria	normal	forte	**sim**
D8	sol	amena	alta	fraca	**não**
D9	sol	fria	normal	fraca	**sim**
D10	chuva	amena	normal	fraca	**sim**
D11	sol	amena	normal	forte	**sim**
D12	nublado	amena	alta	forte	**sim**
D13	nublado	quente	normal	fraca	**sim**
D14	chuva	amena	alta	forte	**não**

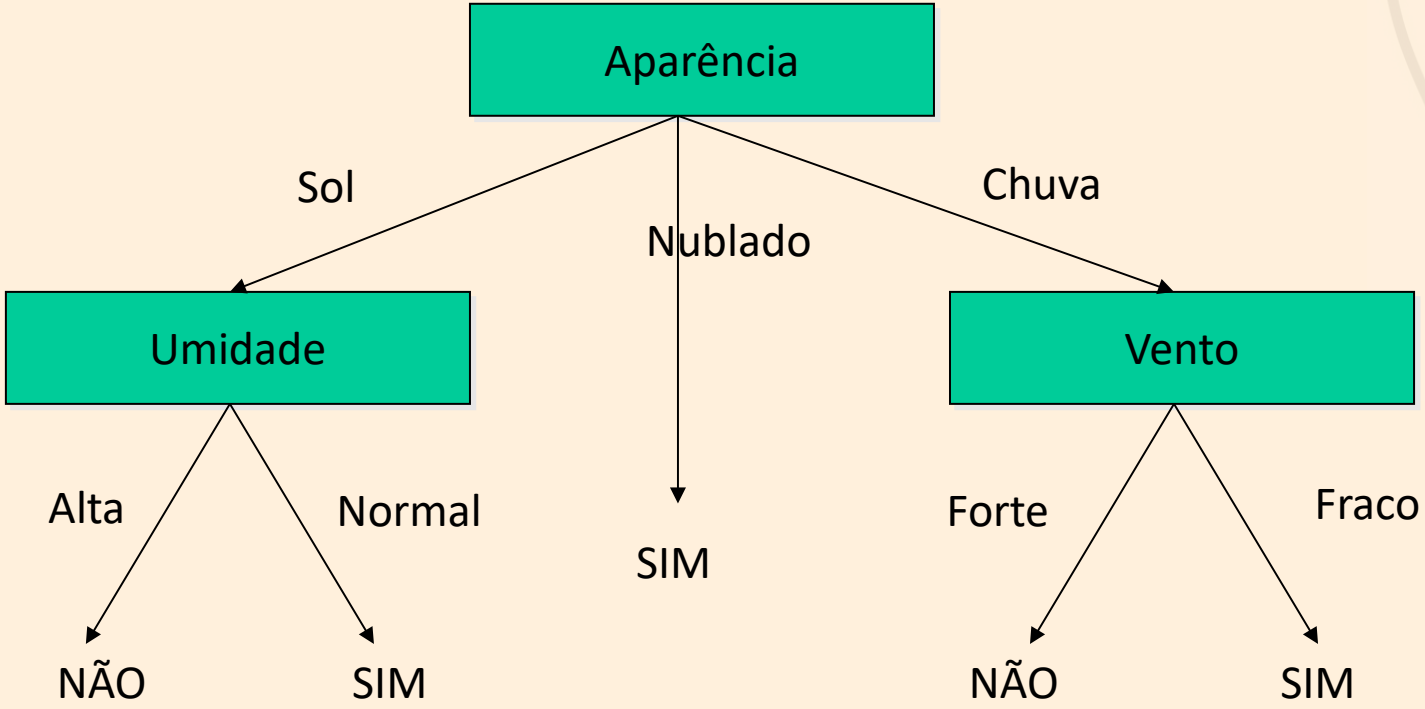
Exercício:

Contemple esta tabela por uns 5 a 10 min e tente obter padrões de que, em que condições climáticas pode haver ou não o jogo.

Ex. basta estar nublado para ter jogo.
(é um padrão que só depende de um atributo, a aparência)

Prof. Waldemar B. Jr.

Uma árvore de decisão mostra as regras que descrevem este padrão



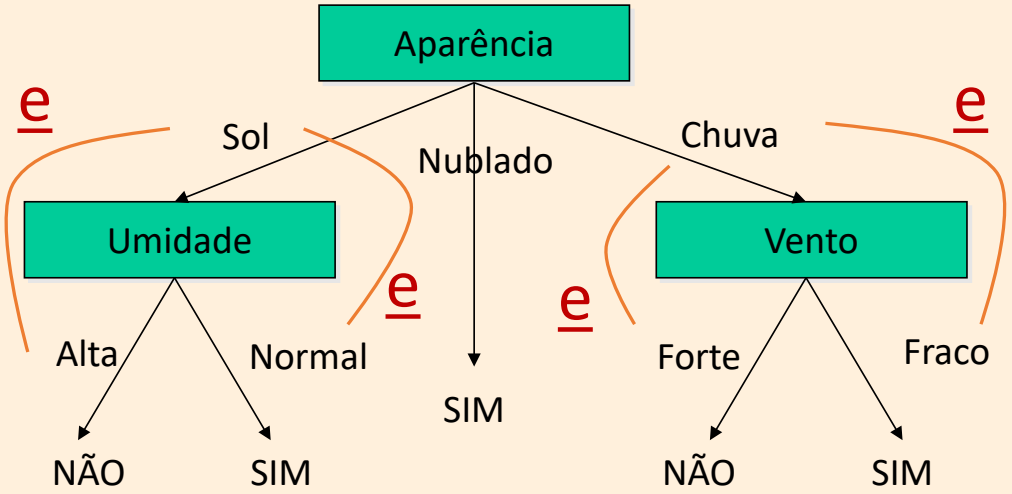
Regras de decisão, a partir da árvore

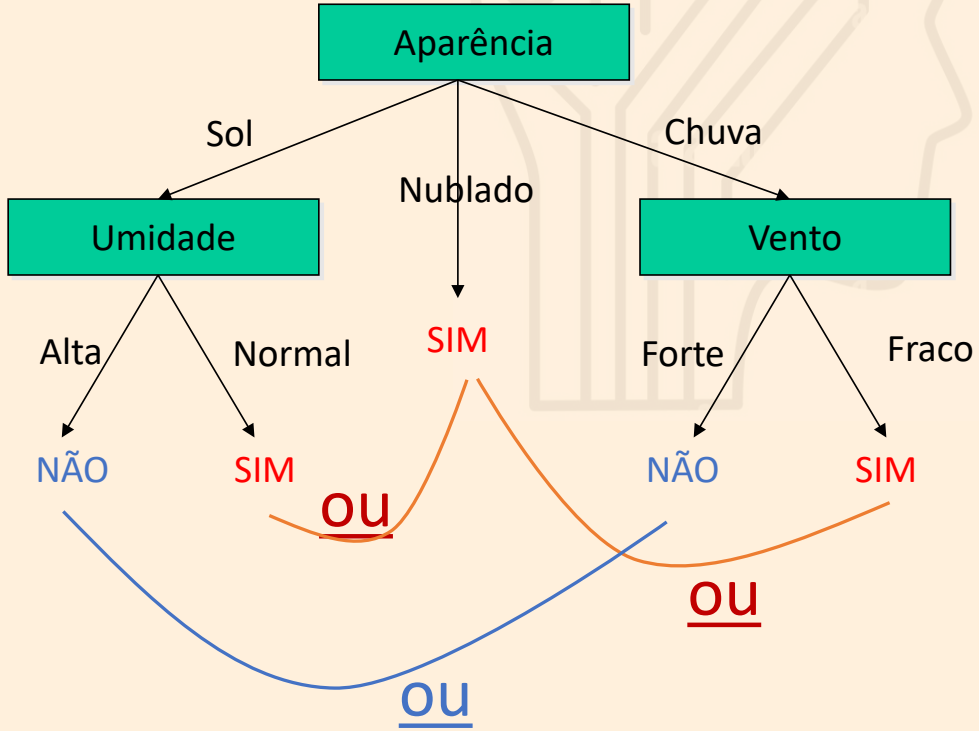
- Se (Aparência == ‘sol’) e (Umidade == ‘alta’) então Jogar = ‘NÃO’
- Se (Aparência == ‘sol’) e (Umidade == ‘normal’) então Jogar = ‘SIM’

- Se (Aparência == ‘nublado’) então Jogar = ‘SIM’

- Se (Aparência == ‘chuva’) e (Vento == ‘forte’) então Jogar = ‘NÃO’
- Se (Aparência == ‘chuva’) e (Vento == ‘fraco’) então Jogar = ‘SIM’

Reescrevendo as regras: lógica aprimorada



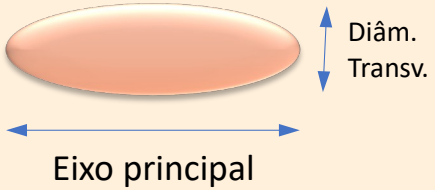


Reescrevendo as regras: lógica aprimorada

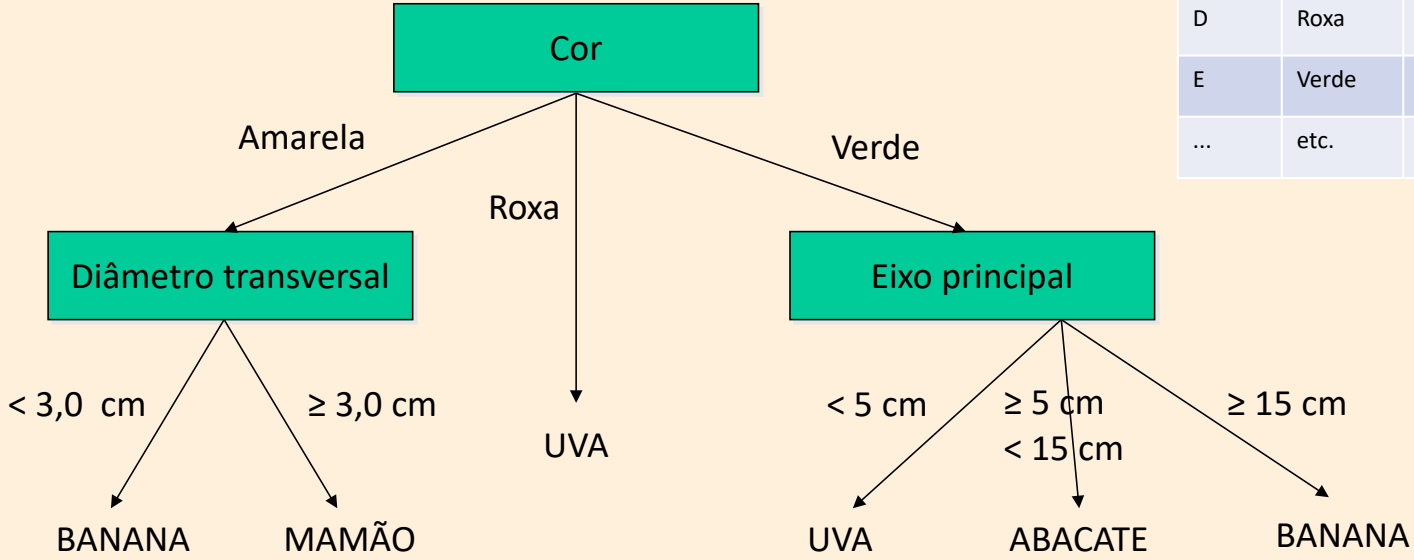
- Se (Aparência == ‘sol’) e (Umidade == ‘normal’)
  ou (Aparência == ‘nublado’)
  ou (Aparência == ‘chuva’) e (Vento == ‘fraco’)
  então Jogar = ‘SIM’

- Se (Aparência == ‘sol’) e (Umidade == ‘alta’)
  ou (Aparência == ‘chuva’) e (Vento == ‘forte’)
  então Jogar = ‘NÃO’

Outro exemplo:
Dados categorizados em mais classes: 26 frutas



FRUTA	COR	EIXO PRINC	DIÂM TRAN	TIPO
A	Verde	12,3 cm	2,8 cm	BANANA
B	Verde	2,3 cm	2,2 cm	UVA
C	Amarela	12,7 cm	2,3 cm	BANANA
D	Roxa	2,3 cm	1,9 cm	UVA
E	Verde	10,1 cm	7,4 cm	ABACATE
...	etc.	etc.	etc.	etc.



Regras de decisão, a partir da árvore (frutas)

- Se (Cor == ‘amarela’) e (Diam.transv. < 3,0) então Fruta = ‘BANANA’
- Se (Cor == ‘amarela’) e (Diam.transv. ≥ 3,0) então Fruta = ‘MAMÃO’

- Se (Cor == ‘roxa’) então Fruta = ‘UVA’

- Se (Cor == ‘verde’) e (Eixo Princ < 5,0) então Fruta = ‘UVA’
- Se (Cor == ‘verde’) e (Eixo Princ ≥ 5,0) e (Eixo Princ < 15,0) então Fruta = ‘ABACATE’
- Se (Cor == ‘verde’) e (Eixo Princ ≥ 15,0) então Fruta = ‘BANANA’

Usando a técnica anterior, de agrupar as regras pela classe Y:

- Se (Cor == ‘amarela’) e (Diam.transv. < 3,0)
  ou (Cor == ‘verde’) e (Eixo Princ ≥ 15,0)
  então Fruta = ‘BANANA’

- Se (Cor == ‘amarela’) e (Diam.transv. ≥ 3,0) então Fruta = ‘MAMÃO’

- Se (Cor == ‘roxa’) ou (Cor == ‘verde’)
  e (Eixo Princ < 5,0)
  então Fruta = ‘UVA’

- Se (Cor == ‘verde’) e (Eixo Princ ≥ 5,0) e (Eixo Princ < 15,0)
  então Fruta = ‘ABACATE’

Não esqueça das precedências lógicas: **e** é processado antes de **ou** EM QUALQUER LINGUAGEM

Resumo

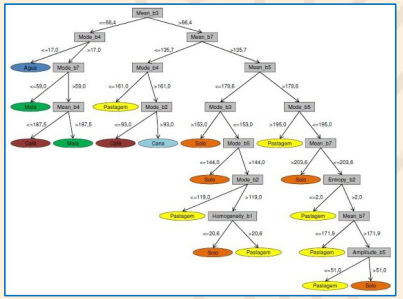
DADOS

FRUTA	COR	EIXO PRINC	DIÂM TRAN	TIPO
A	Verde	12,3 cm	2,8 cm	BANANA
B	Verde	2,3 cm	2,2 cm	UVA
C	Amarela	12,7 cm	2,3 cm	BANANA
D	Roxa	2,3 cm	1,9 cm	UVA
E	Verde	10,1 cm	7,4 cm	ABACATE
...	etc.	etc.	etc.	etc.

Dia	*Aparência*	*Temperatura*	*Umidade*	*Vento*	*Jogar tênis?*
D1	sol	quente	alta	fraca	**não**
D2	sol	quente	alta	forte	**não**
D3	nublado	quente	alta	fraca	**sim**
D4	chuva	amena	alta	fraca	**sim**
D5	chuva	fria	normal	fraca	**sim**
D6	chuva	fria	normal	forte	**não**

AL
GO
RIT
MO

ÁRVORE

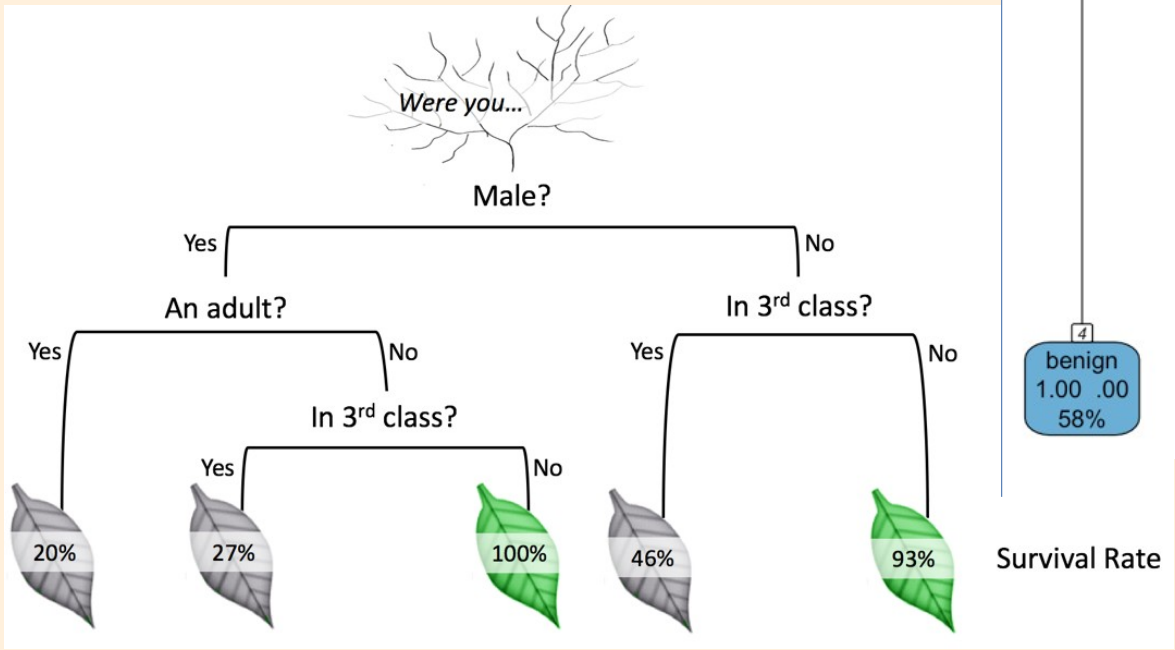


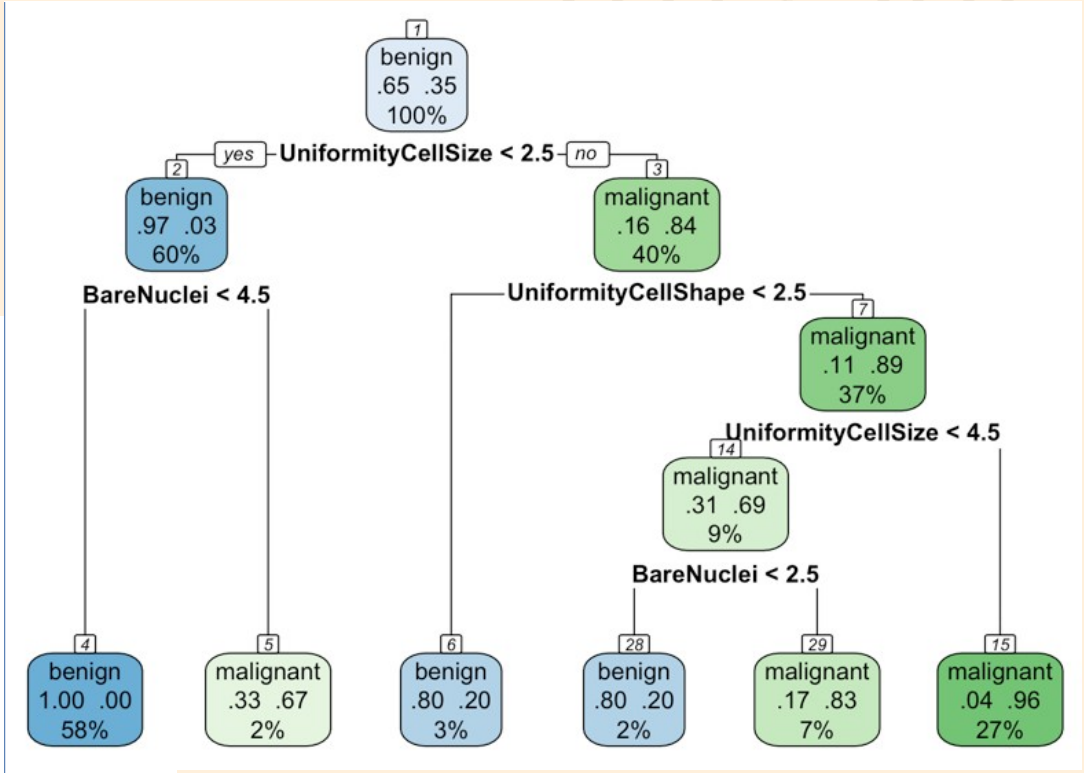
REGRAS LÓGICAS

- SE Tipo de Acidente = Atropelamento de Animal E Modelo da Pista = Reta E Tipo de Veículo = Automóvel E Período do Dia = Noite ENTÃO Causa do Acidente = Animais na Pista. (#Cob = 1036; #Incorr = 17)
- SE Tipo de Acidente = Incêndio E Estado Físico da Pessoa = Ileso E Modelo da Pista = Reta ENTÃO Causa do Acidente = Defeito Mecânico no Veículo. (#Cob = 628; #Incorr = 7)
- SE Tipo de Acidente = Incêndio E Dia da Semana = Quarta-feira E Período do Dia = Manhã ENTÃO Causa do Acidente = Motorista Dormindo. (#Cob = 16)
- SE Tipo de Acidente = Atropelamento de Pessoa E Modelo da Pista = Reta E Período do Dia = Tarde e Tipo de Veículo = Automóvel ENTÃO Causa do Acidente = Falta de Atenção. (#Cob = 343; #Incorr = 85)

Prof. Waldemar B. Jr.

Outros casos:





Survival Rate

Teoria: como é calculado o ganho de informação?

- Qual atributo é o melhor?
  - Selecione o atributo que é mais útil para classificar exemplos.
- Medida quantitativa
  - Ganho de informação
- Para o atributo A, em relação a uma coleção de dados D

$$Gain(D, A) \equiv Entropy(D) - \sum_{v \in Values(A)} \frac{|Dv|}{|D|} Entropy(Dv)$$

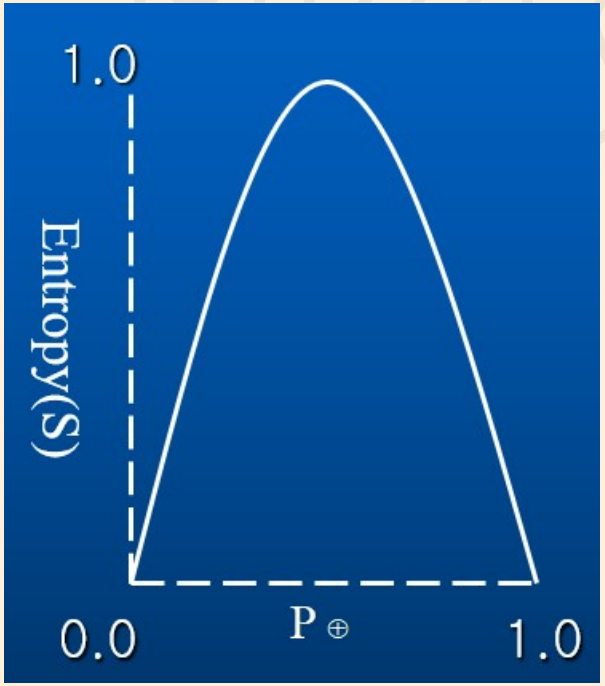
- Redução esperada de entropia

Entropia (informação)

- Impureza de uma coleção arbitrária de exemplos

$$Entropy(D) \equiv \sum_{i=1}^{c} - p_i \log p_i$$

- Número mínimo de bits de informação necessários para codificar a classificação de um membro arbitrário de D



algoritmo

ID3(*Examples*, *Target_attribute*, *Attributes*)

Examples are the training examples. Target_attribute is the attribute whose value is to be predicted by the tree. Attributes is a list of other attributes that may be tested by the learned decision tree. Returns a decision tree that correctly classifies the given Examples.

- Create a *Root* node for the tree
- If all *Examples* are positive, Return the single-node tree *Root*, with label = +
- If all *Examples* are negative, Return the single-node tree *Root*, with label = −
- If *Attributes* is empty, Return the single-node tree *Root*, with label = most common value of *Target_attribute* in *Examples*
- Otherwise Begin
  - $A \leftarrow$ the attribute from *Attributes* that best* classifies *Examples*
  - The decision attribute for *Root* $\leftarrow A$
  - For each possible value, v_i, of A,
    - Add a new tree branch below *Root*, corresponding to the test $A = v_i$
    - Let $Examples_{v_i}$ be the subset of *Examples* that have value v_i for A
    - If $Examples_{v_i}$ is empty
      - Then below this new branch add a leaf node with label = most common value of *Target_attribute* in *Examples*
      - Else below this new branch add the subtree
        ID3($Examples_{v_i}$, *Target_attribute*, *Attributes* − {A}))
- End
- Return *Root*

* The best attribute is the one with highest *information gain*, as defined in Equation (3.4).

Exemplo: jogar tênis (1)

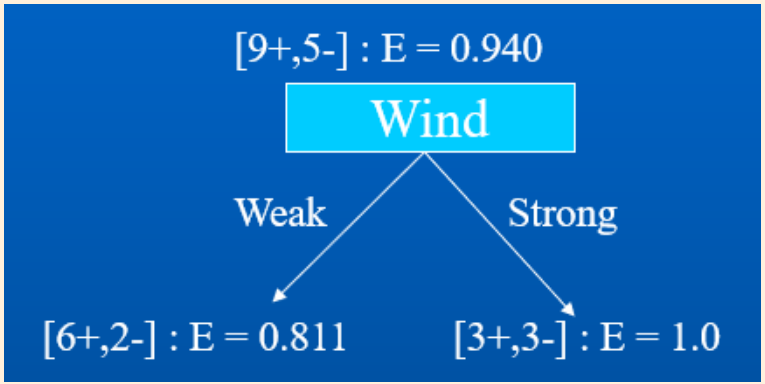
14 dados, sendo 9 positivos (jogar) e 5 negativos (não jogar)

Cálculo da entropia do conjunto

$$Entropy(D) = Entropy([9+,5-])$$
$$= -\frac{9}{14}\log\left(\frac{9}{14}\right) - \frac{5}{14}\log\left(\frac{5}{14}\right)$$
$$= 0.940$$

Exemplo: jogar tênis (2)

Atributo Vento
D = [9+,5-]
D(fraco) = [6+,2-]
D(forte) =[3+,3-]



$$Gain(D,Wind) = Entropy(D) - \sum_{v \in \{weak,strong\}} \frac{|Dv|}{|D|} Entropy(Dv)$$

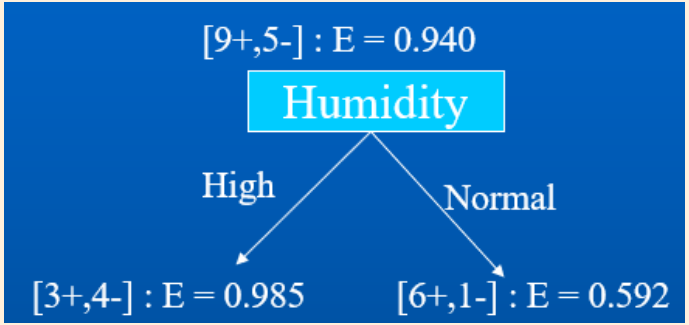
$$= Entropy(D) - \frac{8}{14} Entropy(D_{weak}) - \frac{6}{14} Entropy(D_{strong})$$

$$= 0.940 - \frac{8}{14} 0.811 - \frac{6}{14} 1.00$$

$$= 0.048$$

Exemplo: jogar tênis (3)

Atributo Umidade
D(alta) = [3+,4-]
D(normal) = [6+,1-]



$$Gain(D,Wind) = Entropy(D) - \sum_{v \in \{high,normal\}} \frac{|Dv|}{|D|} Entropy(Dv)$$

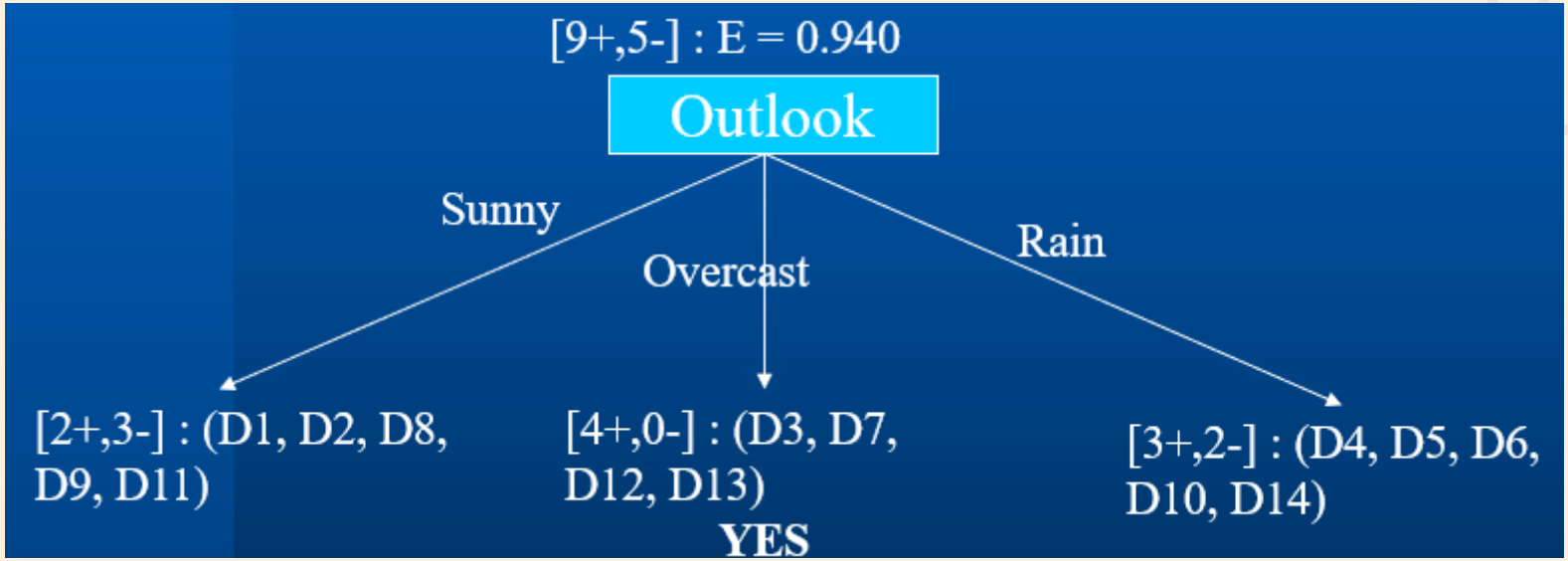
$$= Entropy(D) - \frac{7}{14} Entropy(D_{high}) - \frac{7}{14} Entropy(D_{normal})$$

$$= 0.940 - \frac{7}{14} 0.985 - \frac{7}{14} 0.592$$

$$= 0.151$$

Exemplo: jogar tênis (4)

Melhor atributo?
Ganho (D, Outlook) = 0,246
Ganho(D, Umidade) = 0,151
Ganho(D, Vento) = 0,048
Ganho(D, Temperatura) = 0,029



Exemplo: jogar tênis (5)

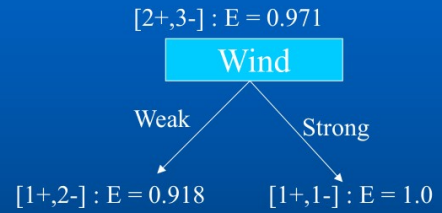
Continuando próximo nível da árvore

$$Entropy(D_{sunny}) = Entropy([2+,3-])$$
$$= -\frac{2}{5}\log\left(\frac{2}{5}\right) - \frac{3}{5}\log\left(\frac{3}{5}\right)$$
$$= 0.971$$

Day	*Outlook*	*Temperature*	*Humidity*	*Wind*	*PlayTennis*
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes

Example : Play Tennis (6)

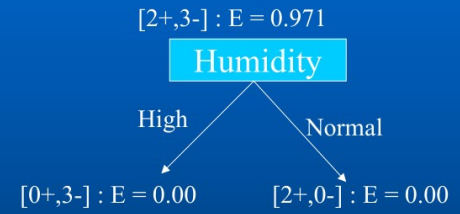
- Attribute *Wind*
  - D_{weak} = [1+,2-]
  - D_{strong}=[1+,1-]



$$
\begin{aligned}
Gain(D,Wind) &= Entropy(D_{sunny}) - \sum_{v\in\{weak,strong\}} \frac{|Dv|}{|D|} Entropy(Dv) \\
&= Entropy(D_{sunny}) - \frac{3}{5} Entropy(D_{weak}) - \frac{2}{5} Entropy(D_{strong}) \\
&= 0.971 - \frac{3}{5} 0.918 - \frac{2}{5} 1.00 \\
&= 0.020
\end{aligned}
$$

Example : Play Tennis (7)

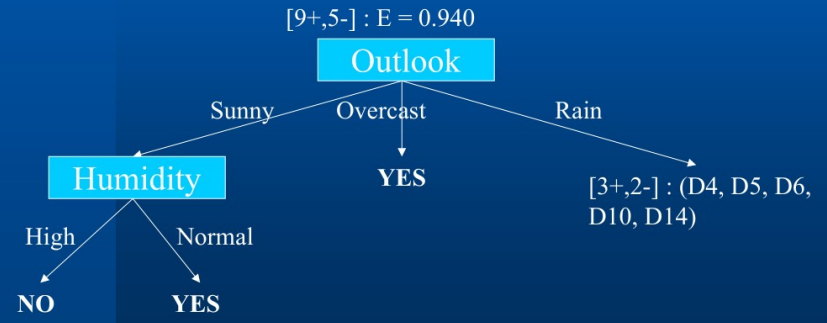
- Attribute *Humidity*
  - D_{high} = [0+,3-]
  - D_{normal}=[2+,0-]



$$
\begin{aligned}
Gain(D,Wind) &= Entropy(D_{sunny}) - \sum_{v\in\{high,normal\}} \frac{|Dv|}{|D|} Entropy(Dv) \\
&= Entropy(D_{sunny}) - \frac{3}{5} Entropy(D_{high}) - \frac{2}{5} Entropy(D_{normal}) \\
&= 0.971 - \frac{3}{5} 0.00 - \frac{2}{5} 0.00 \\
&= 0.971
\end{aligned}
$$

Example : Play Tennis (8)

- Best Attribute?
  - ***Gain(D, Humidity) = 0.971***
  - *Gain*(*D*, *Wind*) = 0.020
  - *Gain*(*D*, *Temperature*) = 0.571



SNU Center for Bioinformation Technology (CBIT)

Example : Play Tennis (9)

- Entropy D_{rain}

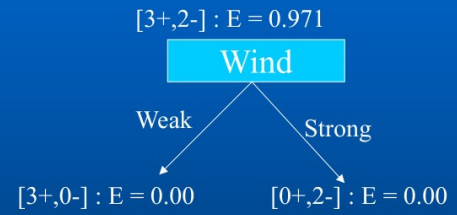
$$Entropy\,(D_{sunny}) = Entropy\,([3+,2-])$$
$$= -\frac{3}{5}\log\left(\frac{3}{5}\right) - \frac{2}{5}\log\left(\frac{2}{5}\right)$$
$$= 0.971$$

Day	*Outlook*	*Temperature*	*Humidity*	*Wind*	*PlayTennis*
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D10	Rain	Mild	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

SNU Center for Bioinformation Technology (CBIT)

Example : Play Tennis (10)

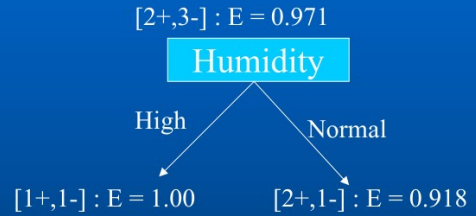
- Attribute *Wind*
  - D_{weak} = [3+,0-]
  - D_{strong}=[0+,2-]



$$Gain(D,Wind) = Entropy(D_{rain}) - \sum_{v\in\{weak,strong\}} \frac{|Dv|}{|D|} Entropy(Dv)$$
$$= Entropy(D_{rain}) - \frac{3}{5} Entropy(D_{weak}) - \frac{2}{5} Entropy(D_{strong})$$
$$= 0.971 - \frac{3}{5}0.00 - \frac{2}{5}1.00$$
$$= 0.971$$

Example : Play Tennis (11)

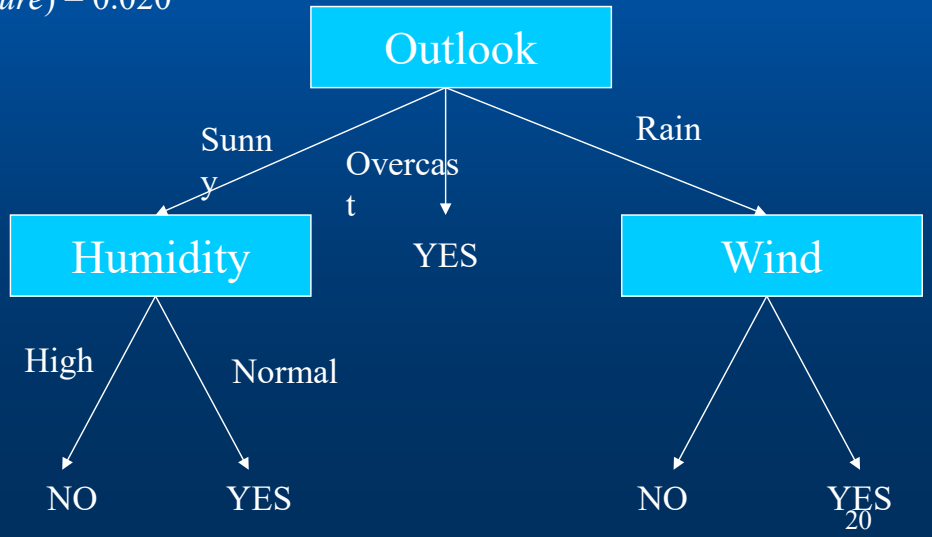
- Attribute *Humidity*
  - D_{high} = [1+,1-]
  - D_{normal}=[2+,1-]



$$Gain(D,Wind) = Entropy(D_{rain}) - \sum_{v\in\{high,normal\}} \frac{|Dv|}{|D|} Entropy(Dv)$$
$$= Entropy(D_{rain}) - \frac{3}{5} Entropy(D_{high}) - \frac{2}{5} Entropy(D_{normal})$$
$$= 0.971 - \frac{2}{5}1.00 - \frac{3}{5}0.918$$
$$= 0.020$$

Example : Play Tennis (12)

- Best Attribute?
  - *Gain*(*D*, *Humidity*) = 0.020
  - ***Gain*(*D*, *Wind*) = 0.971**
  - *Gain*(*D*, *Temperature*) = 0.020

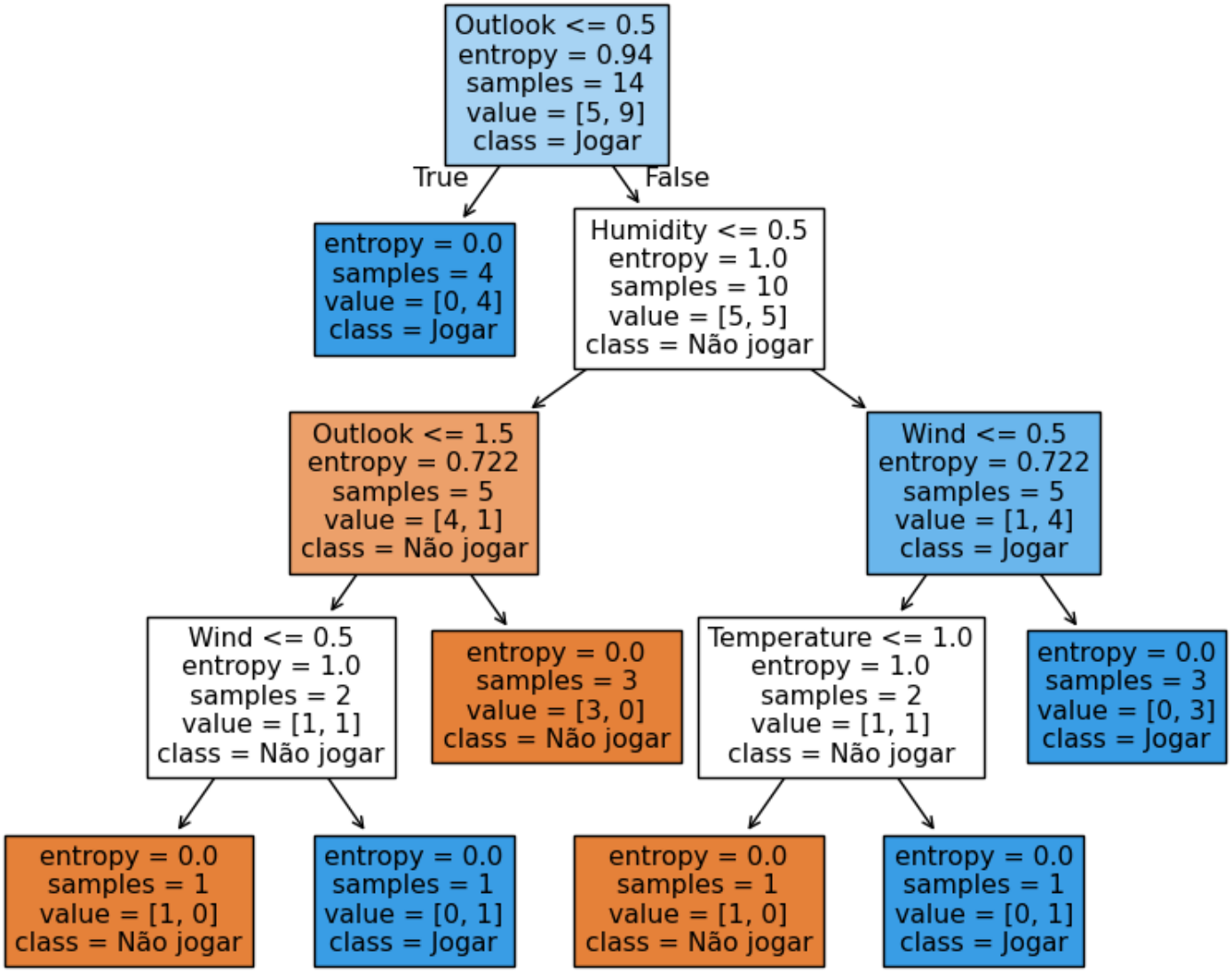


Evitando o superajuste
Considere o seguinte:
(Outlook = Ensolarado e Umidade = Normal e JogarTênis = Não)

Previsão errada da árvore de decisão
(Outlook = Ensolarado e Umidade = Normal) => Sim

E se podarmos o nó "Umidade"?
Quando (outlook = Sunny), PlayTennis => Não

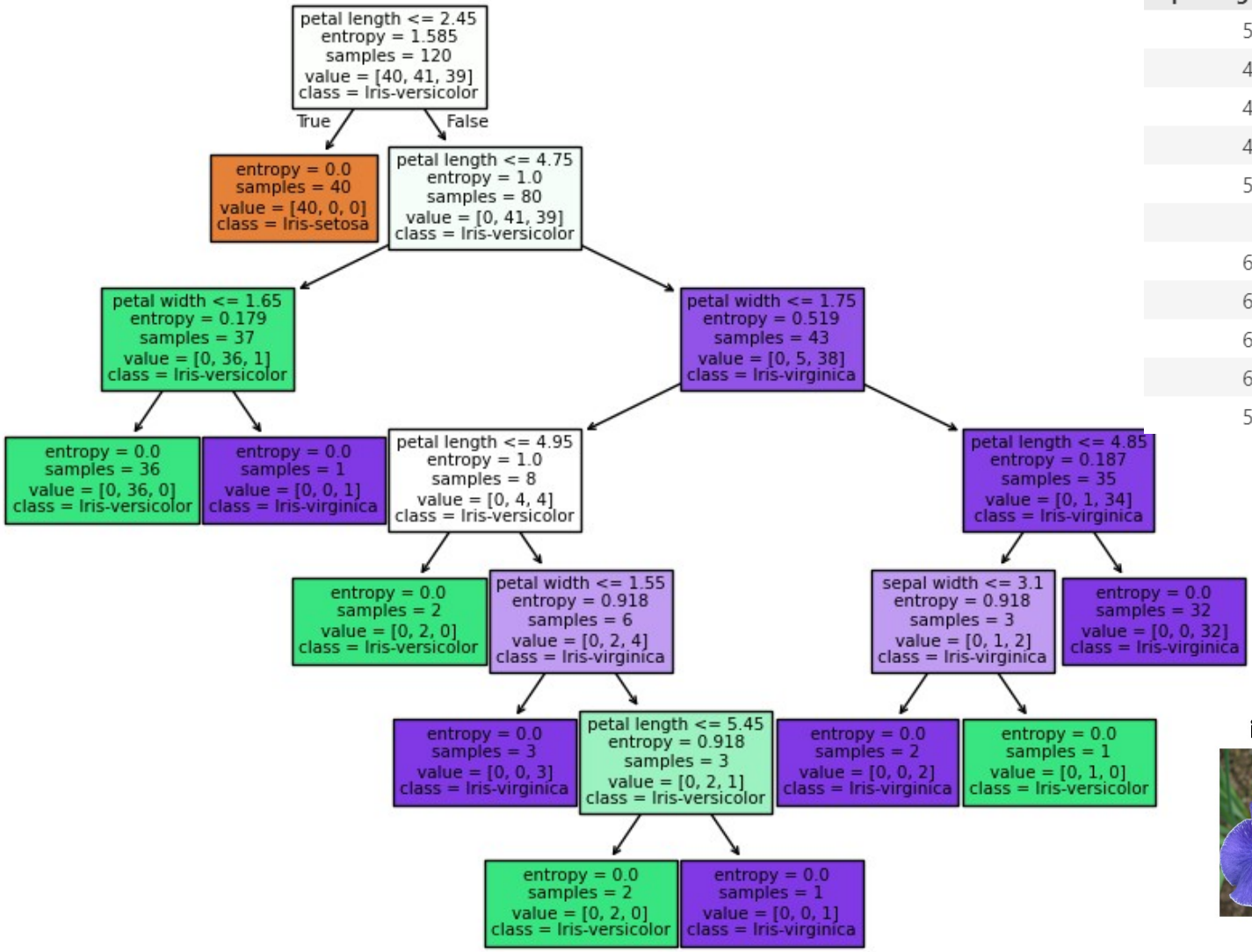
Pode ser previsto corretamente.



Os dados nominais (categóricos) foram convertidos pelo OneHot Encoder
Esses valores é que aparecem nesta árvore

Outlook	Temperature	Humidity	Wind	PlayTennis
Sunny	Hot	High	Weak	No
Sunny	Hot	High	Strong	No
Overcast	Hot	High	Weak	Yes
Rain	Mild	High	Weak	Yes
Rain	Cool	Normal	Weak	Yes

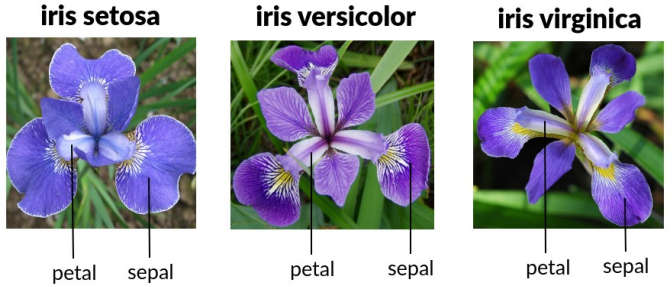
Outlook	Temperature	Humidity	Wind	PlayTennis
2	1	0	1	0
2	1	0	0	0
0	1	0	1	1
1	2	0	1	1
1	0	1	1	1



sepal length	sepal width	petal length	petal width	iris class
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
...	...	...	...	...
6.7	3.0	5.2	2.3	Iris-virginica
6.3	2.5	5.0	1.9	Iris-virginica
6.5	3.0	5.2	2.0	Iris-virginica
6.2	3.4	5.4	2.3	Iris-virginica
5.9	3.0	5.1	1.8	Iris-virginica

Flores “Iris”

https://archive.ics.uci.edu/dataset/53/iris

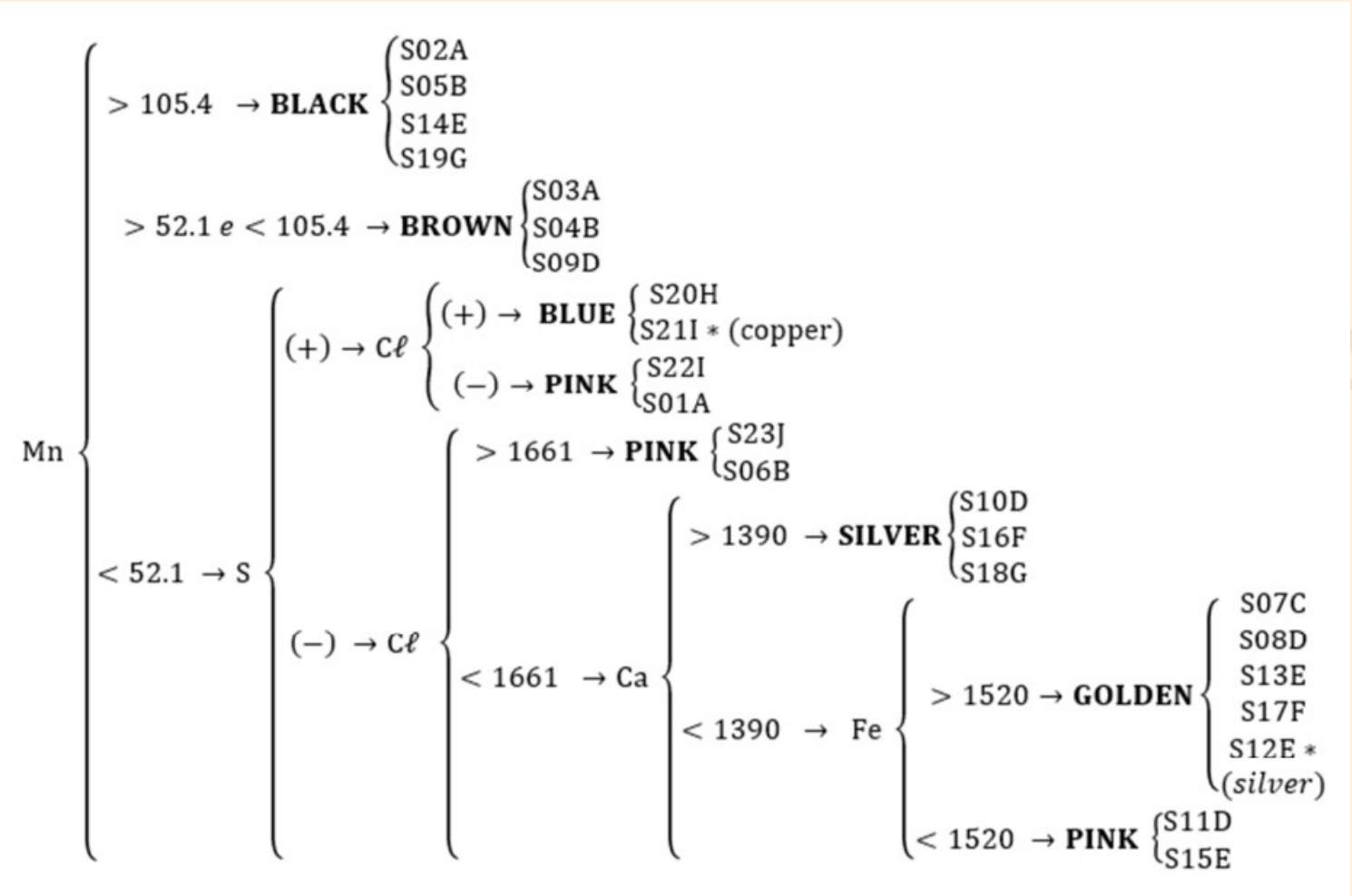


Journal: X-Ray Spectrometry

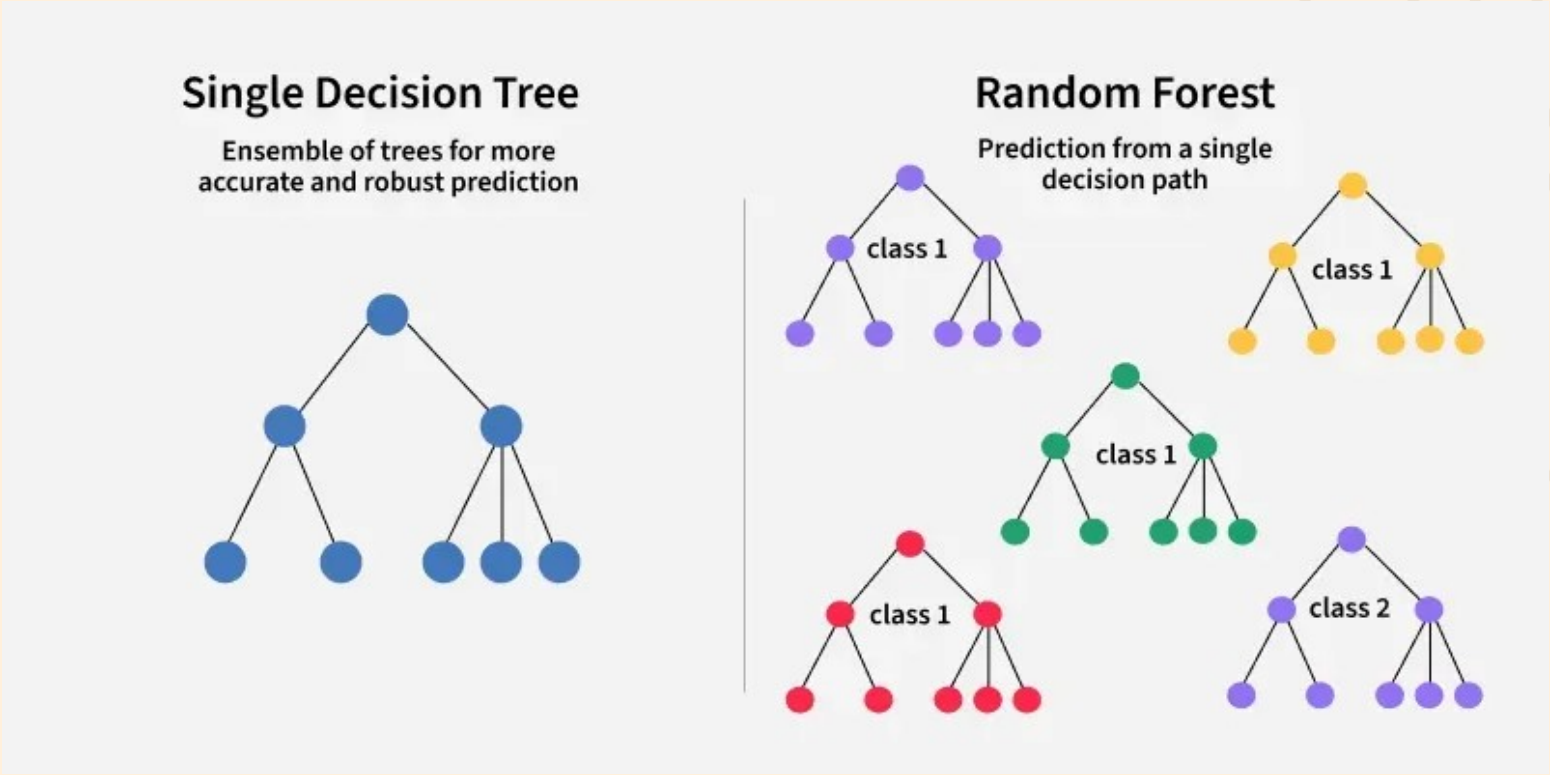
ASSESSMENT OF CHEMICAL ELEMENTS IN COSMETICS' EYESHADOWS BY X-RAY FLUORESCENCE AND INTERNATIONAL NOMENCLATURE OF COSMETIC INGREDIENTS CHARACTERIZATION

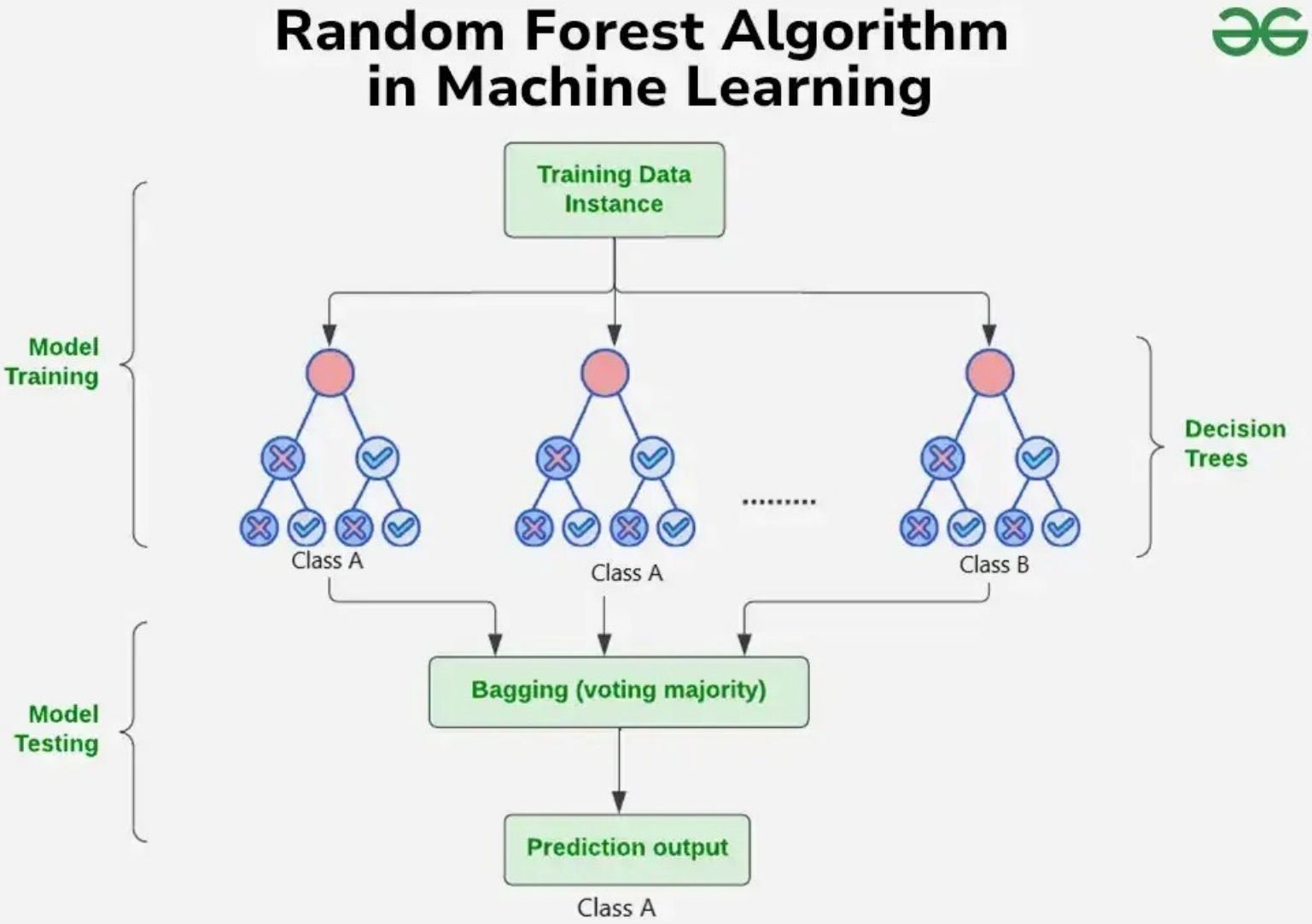
2018

Santos, B.; Oliveira Jr., J.; **Bonventi** Jr., W.; Hanai-Yoshida, W.



Random Forest (árvores aleatórias)



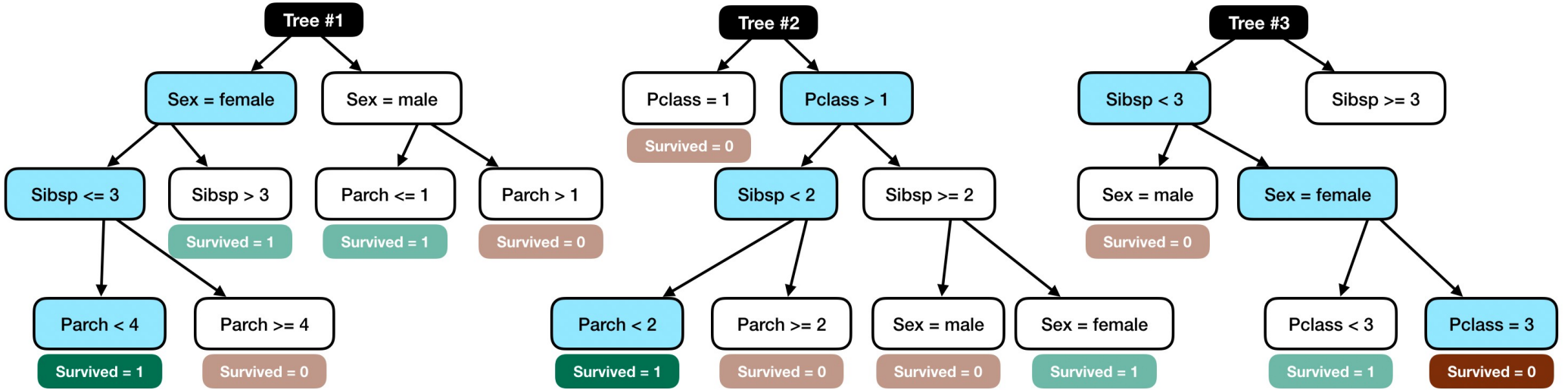

Random Forest Algorithm in Machine Learning
Training Data Instance
Model Training
Decision Trees
Class A
Class A
.....
Class B
Model Testing
Bagging (voting majority)
Prediction output
Class A

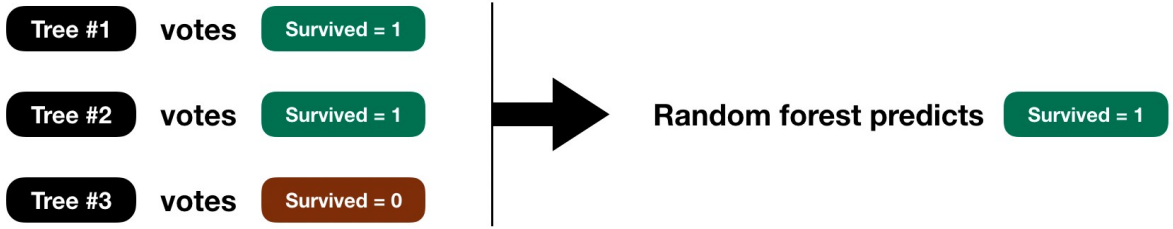
Random Forest

Titanic dataset

Did the passenger survive?

PassengerId	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
893	3	Wilkes, Mrs. James (Ellen Needs)	female	47	1	0	363272	7		S





Evasão universitária

revista Abakós PUC-MG – em revisão

Figura 9 – Principais atributos segundo algoritmo Random Forest.

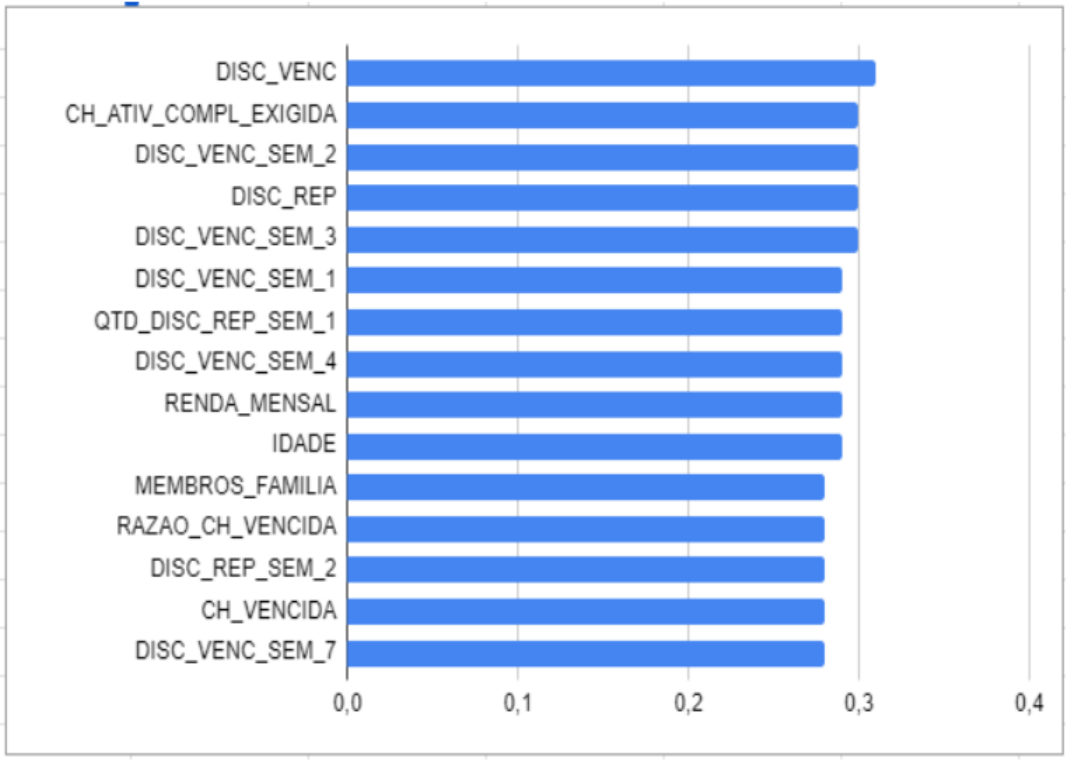


Figura 10 – Atributos menos relevantes