

# Aprendizado de Máquina

Parte I

# Aprendizado de Máquina

```
graph TD; A[Aprendizado de Máquina] --> B[Aprendizado Supervisionado]; A --> C[Aprendizado Não supervisionado]; A --> D[Aprendizado por Reforço];
```

Aprendizado  
Supervisionado

Aprendizado  
Não  
supervisionado

Aprendizado  
por Reforço

# Aprendizado Supervisionado

- Ocorre quando a máquina utiliza dados com atributos-alvo conhecidos (rótulos) para aprender ou treinar.
- A efetividade do aprendizado é validado através de um novo conjunto de dados (teste).
- O modelo treinado utilizada a análise realizada com dados de treinamento para produzir resultados corretos para novos dados avaliados.
- Duas categorias de algoritmos:
  - Classificação: classificam os objetos em determinada classe.
  - Regression: retornam um valor real para objeto fornecido.

# Aprendizado Supervisionado

- Vantagens:
  - Otimiza o desempenho da predição com a ajuda da experiência adquirida.
  - Útil na resolução de vários problemas do mundo real que envolvem problemas de classificação e regressão.
  - Permite controlar o número de classes a partir da definição do conjunto de treinamento.
- Desvantagens:
  - Classificação em big data pode ser complexa.
  - Tem como pré-condição a necessidade de objetos rotulado.
  - Requer uma fase de treinamento.
  - Fase de treinamento demanda tempo computacional.

# Aprendizado Não Supervisionado

- O modelo de aprendizado de máquina aprende usando objetos sem um atributo-alvo relacionado ou conhecido.
- Busca por estruturas que estão presentes no conjunto de dados não rotulados e tais estruturas não são conhecidas.
- O modelo tem liberdade para agrupar os dados em busca de diferenças, similaridades ou padrões sem ter recebido um treinamento prévio.
- Duas categorias de algoritmos:
  - Clustering: busca encontrar agrupamentos a partir dos dados como, por exemplo, agrupar investidores pelo seu comportamento na hora de investir.
  - Association: busca encontrar regras capazes de descrever relações entre os dados, por exemplo, quem compra papinha para bebês também compra fraldas com certa frequência.

# Aprendizado Não Supervisionado

- Vantagens:
  - Não há fase de treinamento e necessidade de objetos com atributo-alvo.
  - Mais fácil de aplicar em situações onde deve haver redução de dimensionalidade
  - Pode retornar padrões desconhecidos existentes nos dados.
  - Usualmente apresenta baixo custo computacional já que não processa dados rotulados.
- Desvantagens:
  - Definir a acurácia ou efetividade sem rótulos ou respostas pode ser um problema.
  - O tomador de decisão acaba precisando interpretar e rotular os dados a partir da classificação obtida.
  - A falta de rótulos pode dificultar a interpretação da relevância de novos padrões encontrados.
  - Dificuldade para lidar com dados faltantes, ausentes, outliers e ruídos.
  - Pode ser pouco escalável para big data ou algoritmos mais complexos.

Parameters	Supervised machine learning	Unsupervised machine learning
Input Data	Algorithms are trained using labeled data.	Algorithms are used against data that is not labeled
Computational Complexity	Simpler method	Computationally complex
Accuracy	Highly accurate	Less accurate
No. of classes	No. of classes is known	No. of classes is not known
Data Analysis	Uses offline analysis	Uses real-time analysis of data
Algorithms used	Linear and Logistics regression, Random forest, Support Vector Machine, Neural Network, etc.	K-Means clustering, Hierarchical clustering, Apriori algorithm, etc.
Output	Desired output is given.	Desired output is not given.
Training data	Use training data to infer model.	No training data is used.
Complex model	It is not possible to learn larger and more complex models than with supervised learning.	It is possible to learn larger and more complex models with unsupervised learning.
Model	We can test our model.	We can not test our model.
Called as	Supervised learning is also called classification.	Unsupervised learning is also called clustering.
Example	Example: Optical character recognition.	Example: Find a face in an image.

Fonte: GeeksforGeeks  
<https://www.geeksforgeeks.org/supervised-unsupervised-learning/>

# Aprendizado Por Reforço

- Busca escolher a melhor ação capaz de maximizar a recompensa em determinada situação.
- Difere de aprendizado supervisionado por não ter respostas ou rótulos, mas o agente ou modelo ainda assim decide o que fazer ao aprender a partir de experiências prévias.
- Os dados são obtidos, acumulados e processados em sistemas de aprendizado que utilizam a abordagem tentativa e erro.
- Os dados não são parte da entrada do modelo que aprende a partir das saídas produzidas em um momento para decidir o que fazer no momento seguinte.
- Após cada ação, o modelo recebe um feedback que permite decidir se a escolha ou ação foi correta, neutra ou incorreta.



<b>Reinforcement learning</b>	<b>Supervised learning</b>
Reinforcement learning is all about making decisions sequentially. In simple words, we can say that the output depends on the state of the current input and the next input depends on the output of the previous input	In Supervised learning, the decision is made on the initial input or the input given at the start
In Reinforcement learning decision is dependent, So we give labels to sequences of dependent decisions	In supervised learning the decisions are independent of each other so labels are given to each decision.
Example: Chess game, text summarization	Example: Object recognition, spam detection

Fonte: GeeksforGeeks <https://www.geeksforgeeks.org/supervised-unsupervised-learning/>

# Aprendizado Por Reforço

- Vantagens:
  - Solucionar problemas complexos.
  - Corrigir possíveis erros durante o aprendizado ou treinamento.
  - Os dados de treinamento são obtidos diretamente da interação modelo e ambiente.
  - Conseguir lidar bem com ambientes incertos, ou seja, onde as saídas nem sempre podem ser previstas.
- Desvantagens:
  - Não é a melhor abordagem para problemas mais simples.
  - Demanda a geração de grande quantidade de dados e processamento.
  - O desempenho depende ou pode ser afetado pela função de recompensa utilizada.
  - Nem sempre é possível saber o motivo do modelo ou agente proceder de determinada forma, dificultando a solução de possíveis problemas.

# Aprendizado Supervisionado

Métodos Baseados em Distância, Árvore de Decisão

# Métodos Baseados em Distâncias

- Hipóteses:
  - Dados similares tendem a estar concentrados em uma mesma região no espaço de entrada.
  - Dados não similares estarão distantes entre si.
- Algoritmos abordados:
  - 1-Vizinho Mais Próximo.
  - K-NN

# Algoritmo do 1-Vizinho Mais Próximo

- Classifica um novo objeto a partir de um conjunto de treinamento, aprendendo com os objetos desse conjunto que estão mais próximos do novo objeto.
- Memoriza apenas os dados de treinamento, logo, torna-se um algoritmo preguiçoso.

# Algoritmo do 1-Vizinho Mais Próximo

- Um objeto é representado como um ponto no espaço de atributos dentro do qual se calculam distâncias entre dois pontos.
- A métrica mais aplicada é a distância euclidiana

$$X = \begin{bmatrix} x_{11} & x_{12} & \dots & x_{1d} \\ x_{21} & x_{22} & \dots & x_{2d} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & \dots & x_{nd} \end{bmatrix}$$

$$\Rightarrow x_u = [x_{u1}, x_{u2}, \dots, x_{ud}]$$

$$x_v = [x_{v1}, x_{v2}, \dots, x_{vd}]$$

$$d(x_u, x_v) = \sqrt{\sum_{j=1}^d (x_{uj} - x_{vj})^2}$$

# Algoritmo do 1-Vizinho Mais Próximo

Conjunto de treinamento:

$$X^{train} = [X|y]$$

$$= \begin{bmatrix} x_{11} & x_{12} & \dots & x_{1d} & y_1 \\ x_{21} & x_{22} & \dots & x_{2d} & y_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ x_{n1} & x_{n2} & \dots & x_{nd} & y_n \end{bmatrix}$$

Objeto de teste:

$$x_t = [x_{t1}, x_{t2}, \dots, x_{td}, y_t]$$

$$d_{min} = +\infty$$

**for**  $i = 1..n$  **do**

**if**  $d(x_i, x_t) < d_{min}$  **then**

$$d_{min} = d(x_i, x_t)$$

$$index = i$$

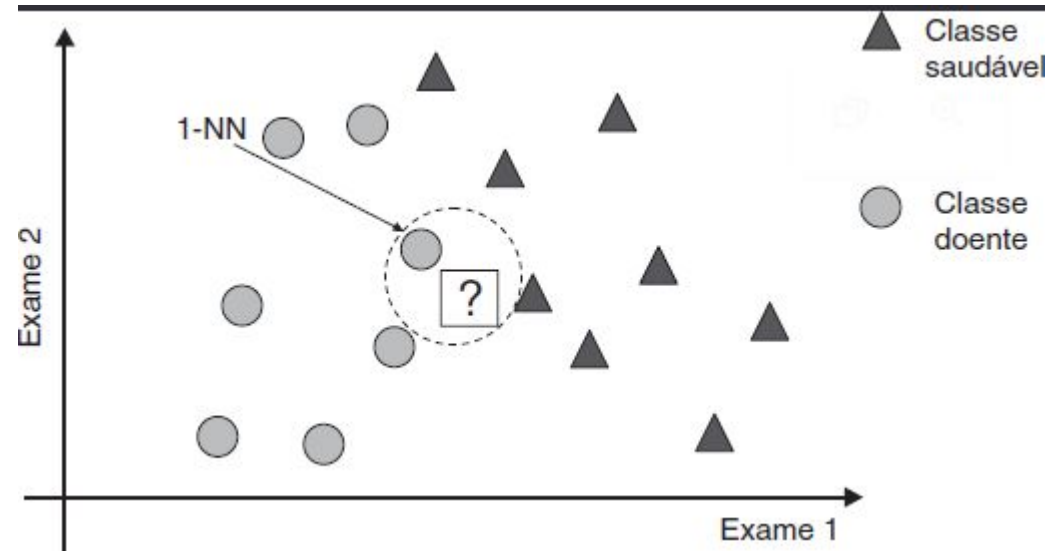
**end if**

**end for**

**return**  $y = y_{index}$

# Algoritmo do 1-Vizinho Mais Próximo

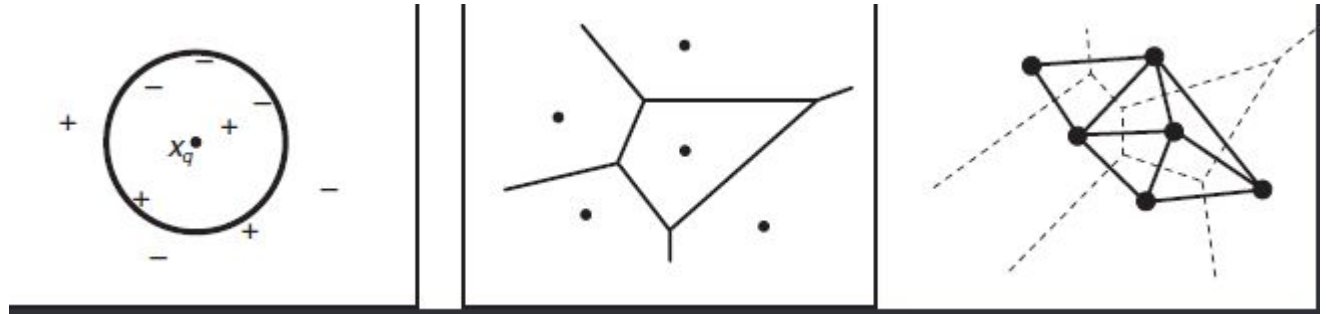
```
 $d_{min} = +\infty$   
for  $i = 1 \dots n$  do  
  if  $d(x_i, x_t) < d_{min}$  then  
     $d_{min} = d(x_i, x_t)$   
     $index = i$   
  end if  
end for  
return  $y = y_{index}$ 
```





# Algoritmo do 1-Vizinho Mais Próximo

- A **superfície de decisão** desenhada pelo algoritmo 1-NN é um conjunto de poliedros convexos contendo cada um dos objetos de treinamento.
- As superfícies de decisão são poliedros convexos com centro em cada objeto do conjunto de treinamento.



# Algoritmo do 1-Vizinho Mais Próximo

- O desempenho dos métodos baseados em distância é afetado pela métrica utilizada.
  - Por exemplo, a distância euclidiana assume atributos numéricos e contínuos.
  - Logo, no caso de atributos qualitativos, deve-se aplicar uma transformação simbólico-numérica.
- A métrica de distância também pode sofrer com variações de escala.
  - Por exemplo, atributos em cm ou km podem afetar o resultado das análises.
  - Logo, transformação numérico-numérico como a normalização pode ser aplicada para minimizar tal problema.

# Algoritmo K-NN

- Extensão do algoritmo 1-NN, onde o termo  $k$  indica a quantidade de vizinhos mais próximos que influenciam no resultado relacionado ao objeto avaliado.
- As previsões dos diferentes vizinhos são agregadas de forma diferente em problemas de classificação e de regressão.

# Algoritmo K-NN

- **Problemas de classificação:** a classe possui um valor dentro de um conjunto discreto, cada vizinho vota em uma classe.
- Classifica-se o objeto avaliado na classe mais votada pelos k vizinhos mais próximo.
- Métrica: moda  $f(x_u) = moda(x_1, x_2, \dots, x_k)$   
 $(x_1, x_2, \dots, x_k) : k$  vizinhos mais próximos de  $x_u$

# Algoritmo K-NN

- **Problemas de regressão:** duas estratégias podem ser utilizadas baseadas na métrica aplicada para determinar os k vizinhos.
- Se a métrica a ser minimizada for o erro quadrático, utiliza-se a média dos valores obtidos para cada um dos vizinhos já que minimiza o erro quadrático.
- Se a métrica a ser minimizada for o desvio absoluto, utiliza-se a mediana já que minimiza o desvio absoluto.

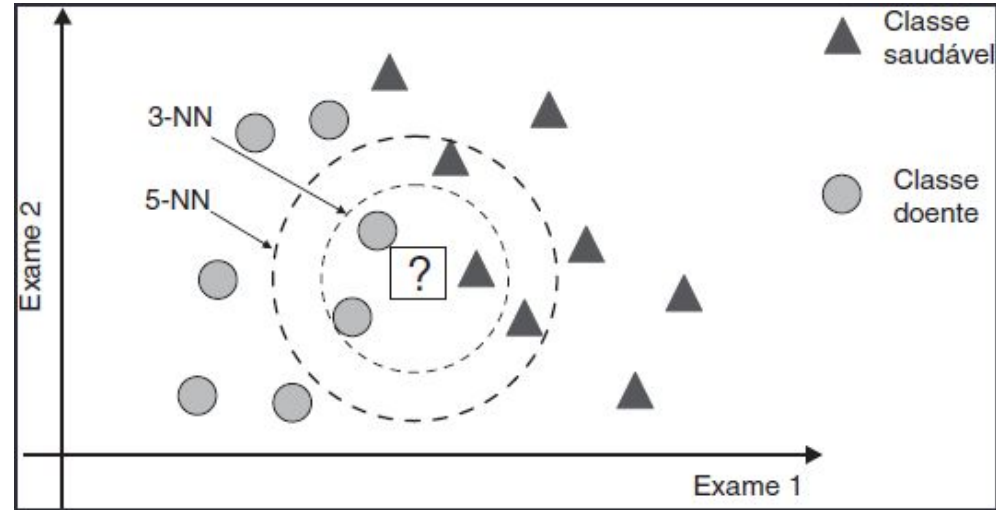
$$f(x_u) = \textit{média}(x_1, x_2, \dots, x_k)$$

$$f(x_u) = \textit{mediana}(x_1, x_2, \dots, x_k)$$

$(x_1, x_2, \dots, x_k)$  : k vizinhos mais próximos de  $x_u$

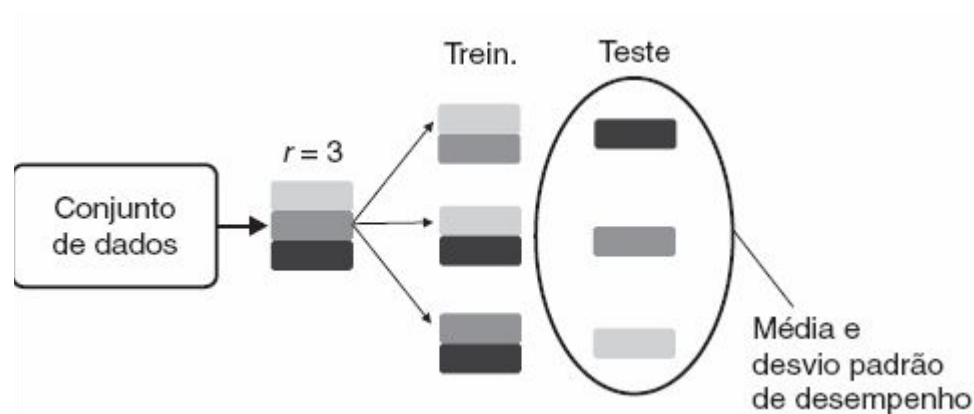
# Algoritmo K-NN

- O parâmetro  $k$  é definido pelo usuário, sendo um número ímpar para evitar empates em problemas de classificação com duas classes.
- Estratégias:
  - Estimar  $k$  por validação cruzada.
  - Associar um peso à contribuição de cada vizinho.



# Algoritmo K-NN

- **Validação Cruzada (r-fold cross-validation):** O conjunto de dados  $X$  é particionado em  $r$  subconjuntos com aproximadamente a mesma cardinalidade, onde a fase de treinamento do preditor utiliza objetos das  $r - 1$  partições, enquanto a partição faltante é utilizada para teste.
- Uma combinação diferente de partições é utilizada num total de  $r$  treinamentos e testes, onde o desempenho final do preditor é obtido pela média dos desempenhos alcançados em cada uma das  $r$  etapas.



# Algoritmo K-NN

- O **r-fold cross-validation estratificado** mantém a proporção de cada classe em cada partição gerada.
  - Por exemplo, se há uma classe minoritária com 20% dos objetos e uma classe majoritária com 80% dos objetos, essa proporção tende a ser preservada em cada partição.
- A abordagem **leave-one-out** é aplicada quando se decide utilizar  $r=n$ , com um objeto separada para teste e  $n - 1$  objetos usados no treinamento do preditor.
  - Atinge uma estimativa mais apurada a um custo computacional considerável, por isso, aplica-se mais a uma amostra de dados pequena.



# Algoritmo K-NN

- Problema com a validação cruzada: parte dos dados é compartilhada entre os subconjuntos de treinamento.
- Para  $r \geq 2$  partições, uma proporção de  $(1-2/r)$  dos objetos é compartilhada (Monard e Baranauskas, 2003).
  - Por exemplo, utilizando 10 partições (folds), 80% dos objetos contidos nos subconjuntos de treinamento são compartilhados. Logo, embora as partições de teste sejam distintas entre si, com  $r > 2$  não se tem completa independência entre os subconjuntos de treinamento.

# Algoritmo K-NN

- Associar um peso à contribuição de cada vizinho: a contribuição de cada um dos k vizinhos é ponderada de forma inversamente proporcional à distância ao ponto de teste.
- Nesse caso, pode-se utilizar todos os objetos do conjunto de treinamento, ou seja,  $k = n$ .
- Utiliza-se a média ponderada para problemas de classificação como abaixo:

$$y_t = \operatorname{argmax}_c \left\{ \sum_{l=1}^k w_l \times I(c, y_l) \right\}$$

$$w_l = \frac{1}{d(x_t, x_l)}$$

$$I(c, y_l) = \begin{cases} 1, & c = y_l \\ 0, & \text{caso contrário} \end{cases}$$

# Algoritmo K-NN

- Vantagens:
  - O algoritmo de treinamento é simples (armazenar objetos apenas).
  - Constrói aproximações locais da função objetivo, diferentes para cada novo dado a ser classificado. Essa característica pode ser vantajosa quando a função objetivo é muito complexa, mas ainda pode ser descrita por uma coleção de aproximações locais de menor complexidade (Mitchell, 1997).
  - Aplicável mesmo em problemas complexos.
  - Naturalmente incremental, ou seja, para novos exemplos de treinamento disponíveis, basta armazená-los na memória.

# Algoritmo K-NN

- Desvantagens:
  - **Predição custosa:** a fase de treinamento requer pouco esforço computacional, mas classificar um novo objeto requer calcular a distância dele a todos os objetos de treinamento.
  - **Dados redundantes:** como todos os algoritmos baseados em distâncias, ele é afetado pela presença de atributos redundantes e de atributos irrelevantes.

# Algoritmo K-NN

- Desvantagens:
  - **Problemas com a dimensionalidade:** uma distância elevada entre pontos pode ser causada pela dimensionalidade.
  - Suponha 100 pontos com distribuição uniforme, por exemplo, representados por um cubo ou outros poliedros com cada lado medindo 1 unidade.

Núm. dimensões	Distância média
2	0,494
3	0,647
4	0,772
5	0,875
...	
10	1,280

# Algoritmo K-NN

- Desvantagens:
  - A densidade diminui e deixa o conjunto de dados esparso.
  - Para 10 dimensões, a distância média é maior que o tamanho do lado do hipercubo!
  - Quando a dimensão aumenta linearmente, para manter a mesma densidade de pontos, é necessário aumentar de forma exponencial o número de pontos.
  - Beyer et al. (1999) mostram que, sob um amplo conjunto de condições (muito mais amplo do que dimensões independentes e identicamente distribuídas), com o aumento da dimensionalidade, a distância ao vizinho mais próximo aproxima-se da distância ao vizinho mais afastado.

# Algoritmo K-NN

- Desvantagens:
  - A dimensionalidade de um problema pode afetar de forma negativa o desempenho dos algoritmos baseados em distâncias.
  - Uma das formas para reduzir o impacto da dimensionalidade de um problema consiste em selecionar um subconjunto de atributos relevantes para o problema tratado.

# Árvores de Decisão

- As árvores de decisão estruturam o conhecimento obtido através de uma hierarquia de decisões.
- Tais decisões são refinadas até a obtenção da classificação final.
- Essa abordagem estruturante permite compreender melhor o processo decisório em um problema de classificação ou regressão.
- A explicabilidade em modelos de aprendizado de máquina é relevante na solução dos problemas práticos, uma vez que se deve para garantir maior transparência nos resultados obtidos.
  - Por exemplo, um crédito negado a um cliente por um modelo de aprendizado deve fornecer os motivos para tal decisão.



# Árvores de Decisão

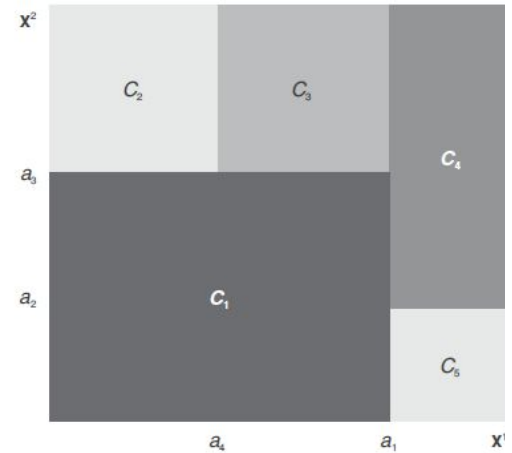
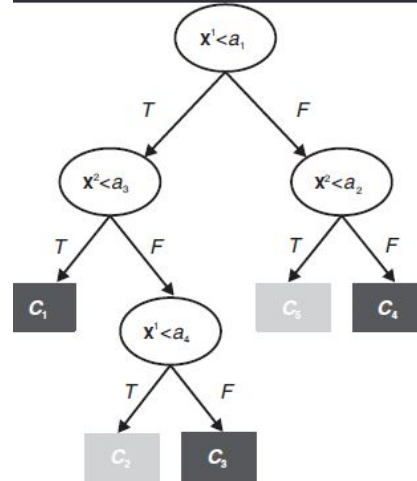
- Abordagem: utilizada a estratégia de divisão e conquista, onde o problema original é dividido em problemas mais simples sobre os quais uma mesma estratégia é aplicada recursivamente.
- Logo, divide-se o espaço de instâncias em subespaços sobre os quais se aplicam diferentes modelos.
- As soluções encontradas para os problemas mais simples são combinadas em uma estrutura de árvore para fornecer a solução do problema original.

# Árvores de Decisão

- Temos **árvores de decisão** para problemas de classificação e **árvores de regressão** para problemas de regressão, mas usaremos o termo **árvore de decisão** de forma genérica.
- Nos dois casos, os algoritmos de indução das árvores e a interpretação dos modelos são parecidos.

# Árvores de Decisão

- Os nós folhas contêm o atributo alvo ou um símbolo nulo indicando que não é possível atribuir nenhum valor ou classe ao nó.
- Os nós internos contêm valores de um atributo ou testes condicionais baseado nos valores dos atributos.
- Os arcos podem conter o teste condicional ou o resultado do teste condicional.



# Árvores de Decisão

- Uma árvore pode ser gerada pela divisão do conjunto de dados em subconjuntos através da Attribute selection measure (ASM) ou Medida de Seleção de Atributos.
- MSA avalia qual atributo criará os subconjuntos de dados mais homogêneos após a divisão, maximizando o ganho de informação.
- A impureza nos dados afeta o ganho de informação.

# Árvores de Decisão

- Impureza:
  - uma medida da homogeneidade do atributo-alvo considerando um subconjunto de dados (atributos).
  - uma medida da homogeneidade dos rótulos em um nó da árvore.
  - Refere-se ao grau de aleatoriedade ou incerteza de um conjunto de objetos em um subconjunto de dados
- Gini impurity e entropia são duas medidas de impureza para problemas de classificação.
- Variância é uma medida de impureza para problemas de regressão.

# Árvores de Decisão

- Entropia: mede o grau de aleatoriedade ou incerteza presente no conjunto de dados.
  - Alta entropia indica um subconjunto de dados mais heterogêneo, por exemplo, com classes mais diversificadas em um problema de classificação.
  - Baixa entropia indica um subconjunto de dados mais homogêneo ou puro.

$$H(Y) = - \sum_{i=1}^k Pr(Y = y_i) \log_2 Pr(Y = y_i)$$

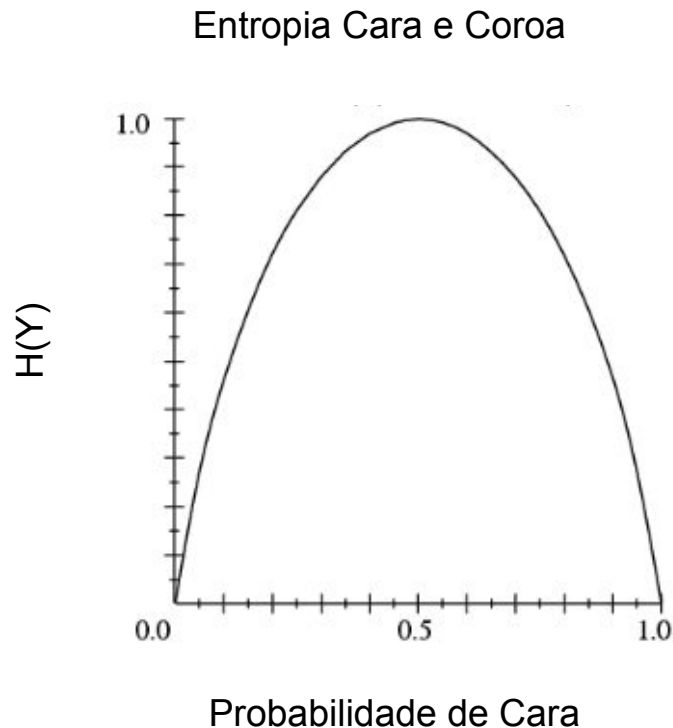
# Árvores de Decisão

Exemplo: Conjunto de dados com resultado de dois lançamentos da moeda em 6 tentativas

$X_1$	$X_2$	Y
V	V	V
V	F	V
V	V	V
V	F	V
F	V	V
F	F	F

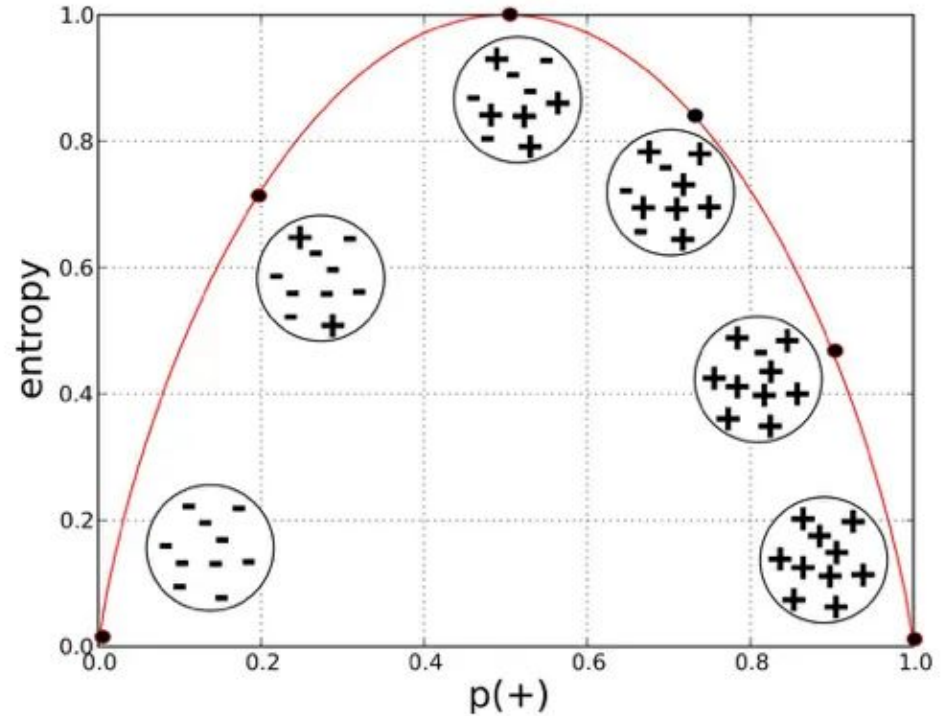
$$Pr(Y = V) = \frac{5}{6} \quad Pr(Y = F) = \frac{1}{6}$$

$$H(Y) = -\frac{5}{6} \log_2 \frac{5}{6} - \frac{1}{6} \log_2 \frac{1}{6} = 0.65$$



# Árvores de Decisão

Exemplo: Considere um conjunto de dados com 100 objetos classificados em apenas duas classes: + e -, onde 30 objetos pertencem à classe positiva e 70 pertencem à classe negativa.



$$H(Y) = -\frac{3}{10} \log_2 \frac{3}{10} - \frac{7}{10} \log_2 \frac{7}{10} = 0.88$$



# Árvores de Decisão

- Entropia Condicional: Sejam X e Y variáveis aleatórias com Y condicionado à X.

$$H(Y|X) = - \sum_{j=1}^v Pr(X = x_j) \sum_{i=1}^k Pr(Y = y_i | X = x_j) \log_2 Pr(Y = y_i | X = x_j)$$

$$Pr(X_1 = V) = \frac{4}{6} \quad Pr(X_1 = F) = \frac{2}{6}$$

$$Pr(Y = V | X_1 = V) = 1 \quad Pr(Y = F | X_1 = V) = 0$$

$$Pr(Y = V | X_1 = F) = \frac{1}{2} \quad Pr(Y = F | X_1 = F) = \frac{1}{2}$$

$$\begin{aligned} H(Y|X_1) &= -\frac{4}{6} \times (1 \times \log_2 1 + 0 \times \log_2 0) \\ &\quad - \frac{2}{6} \left( \frac{1}{2} \times \log_2 \frac{1}{2} + \frac{1}{2} \times \log_2 \frac{1}{2} \right) \\ &= \frac{2}{6} \end{aligned}$$

$X_1$	$X_2$	Y
V	V	V
V	F	V
V	V	V
V	F	V
F	V	V
F	F	F

# Árvores de Decisão

- Information Gain ou ganho de informação: mede a redução de impureza alcançado pela divisão do conjunto de dados usando um atributo em particular na árvore de decisão.
- O critério da divisão é determinado pelo atributo que oferece o maior ganho de informação, ou seja, determina o atributo com melhor informação para realizar a divisão do conjunto de dados em cada nó da árvore buscando gerar subconjuntos o mais puros (homogêneos) possíveis.

# Árvores de Decisão

- Logo, o Information Gain (IG) mede a redução na entropia ou incerteza após o particionamento.
- $IG(X)$ : Maior a redução na incerteza, maior o ganho de informação obtido a respeito de  $Y$  (atributo-alvo) em relação à  $X$  (conjunto de dados).

$$IG(X) = H(Y) - H(Y|X)$$

$$IG(X) = H(Y) - H(Y|X) = 0.65 - 0.33 = 0.32$$

$x_1$	$x_2$	$Y$
V	V	V
V	F	V
V	V	V
V	F	V
F	V	V
F	F	F

# Árvores de Decisão

- O método começa com uma árvore vazia ou nó raiz contendo o conjunto de dados completo.
- O particionamento é aplicada usando o critério de divisão, por exemplo, utilizando  $IG(X)$  como métrica para seleção de atributo:

$$\arg \max_i \{IG(X_i)\} = \arg \max_i \{H(Y) - H(Y|X_i)\}$$

- Recursivamente, aplica-se tal critério aos nós filhos gerados.

# Árvores de Decisão

Passo 1: Nó raiz recebe o conjunto de dados completo (dados de treinamento).

Passo 2: Para cada atributo:

Passo 2.1: Particione o conjunto de dados no nó pelo valor do atributo.

Passo 2.2: Calcule o ganho de informação da partição.

Passo 3: Identifique o atributo que resulta no maior ganho de informação.

Passo 3.1: Ajuste esse atributo como o critério de divisão do nó atual.

Passo 3.2: Se o melhor ganho de informação leva a nó “puro”, o nó atual é um nó folha.

Passo 4: Particione todos as instâncias de acordo com o valor do melhor atributo.

Passo 5: Denote cada partição como um nó filho do nó atual.

Passo 6: Para cada nó filho:

Passo 6.1: Se o nó filho é “puro”, esse nó é um nó folha

Passo 6.2: Senão, o nó filho se torna o nó atual e retorne ao Passo 2.

# Árvores de Decisão

- Vantagens:
  - Segue um procedimento próximo à tomada de decisão humana, tornando o método simples de ser entendido.
  - Útil na solução de problemas de tomada de decisão.
  - Permite inferir sobre os possíveis resultados de um problema.
  - Trata impureza de dados, ou seja, menos dependente da limpeza de dados do que outros métodos.

# Árvores de Decisão

- Desvantagem:
  - Apresenta várias “camadas” o que pode torná-lo complexo.
  - Tendência a gera modelos com overfitting o que pode ser solucionado usando Random Forest.
  - Complexidade computacional aumenta quando há classes com muitos rótulos.

# Árvores de Decisão

- Características dos problemas para aplicar árvore de decisão:
  - Instâncias com pares atributo-valor: Relações como, por exemplo, uma temperatura e seu rótulo (fria, morna, quente) ou valores numéricos em determinada escala.
  - Atributo-alvo com valores discretos distintos: adequado para atributos-alvo booleanos (0 ou 1 / falso ou verdadeiro), mas o conceito se expande além de atributos-alvo duais.



# Árvores de Decisão

- Características dos problemas para aplicar árvore de decisão:
  - Descrições disjuntivas: árvore de decisão lida adequadamente com particionamentos (situação descritas por expressões disjuntivas).
  - Conjuntos de dados com ruídos: alta resiliência com inconsistências de dados ou discrepância nos atributos
  - Conjuntos de treinamento com dados faltantes: modelo pode ser treinado ainda que nem todos os atributos estejam presentes. Por exemplo, aprender sobre o atributo nível de humidade num dia a partir de um conjunto específico de treinamento.

# Árvores de Decisão

- Problemas práticos:
  - Determinar a profundidade da árvore de decisão;
  - Tratar atributos contínuos.
  - Escolher uma medida apropriada para seleção de atributo.
  - Lidar com conjunto de treinamento com valores de atributos faltantes.
  - Lidar com atributos com diferentes custos ou pesos.
  - Melhorar a eficiência computacional