Machine Learning

Parte 2

Evandro J.R. Silva

ejrs.profissional@gmail.com

Bacharelado em Ciência da Computação
Faculdade Estácio Teresina

23 de julho de 2022

Sumário

Principais Algoritmos

Principais Algoritmos

- Aprendizado Supervisionado
- Aprendizado Não Supervisionado

Principais Algoritmos

- Aprendizado Supervisionado
  - Classificação
  - Regressão

Principais Algoritmos

- Aprendizado Supervisionado
  - Classificação
    - Naive Bayes

Estácio

Principais Algoritmos

- Aprendizado Supervisionado
  - Classificação
    - Naive Bayes
    - k-NN

Estácio

Principais Algoritmos

- Aprendizado Supervisionado
  - Classificação
    - Naive Bayes
    - k-NN
    - Árvore de Decisão

Principais Algoritmos

- Aprendizado Supervisionado
  - Classificação
    - Naive Bayes
    - k-NN
    - Árvore de Decisão
    - Redes Neurais Artificiais

Principais Algoritmos

- Aprendizado Supervisionado
  - Regressão

Principais Algoritmos

- Aprendizado Supervisionado
  - Regressão
    - Regressão Logística

Principais Algoritmos

- Aprendizado Supervisionado
- Aprendizado Não Supervisionado

Principais Algoritmos

- Aprendizado Supervisionado

- Aprendizado Não Supervisionado
  - k-Means

Aprendizado Supervisionado

Classificação

Naive Bayes

Naive Bayes

- Naive Bayes é um **modelo de classificação** baseado na **probabilidade condicional**.
- A partir da Teoria da Decisão Bayesiana, podemos calcular, para uma dada observação, qual sua classe mais provável.

Estácio

Naive Bayes

- Naive Bayes é um **modelo de classificação** baseado na **probabilidade condicional**.
- A partir da Teoria da Decisão Bayesiana, podemos calcular, para uma dada observação, qual sua classe mais provável.
  - Classe ou categoria. Daí vem o termo *classificação*. A partir de um ou mais atributos podemos separar objetos em diferentes classes.

Naive Bayes

Lembremos dos peixes:

Figura: Salmão

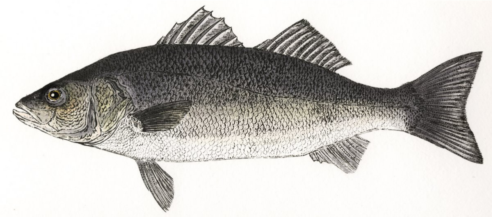


Figura: Robalo

Naive Bayes

- A tarefa: classificar um peixe como Salmão ou Robalo.
  - Classes: $\omega_1 = \textit{Robalo} \qquad \omega_2 = \textit{Salmão}$.

Naive Bayes

- A tarefa: classificar um peixe como Salmão ou Robalo.
  - Classes: $\omega_1 =$ *Robalo* $\omega_2 =$ *Salmão*.
- *A priori* não se sabe a qual espécie o peixe pertence.

Estácio

Naive Bayes

- A tarefa: classificar um peixe como Salmão ou Robalo.
  - Classes: $\omega_1 = \textit{Robalo} \qquad \omega_2 = \textit{Salmão}$.
- *A priori* não se sabe a qual espécie o peixe pertence.
- O brilho do peixe foi observado.

Estácio

Naive Bayes

- Probabilidade *a priori*
  - Reflete o quão **verossímil** é observar uma das duas espécies de peixe.
  - Se a quantidade de Salmão é **igual** à quantidade de Robalo, então é igualmente verossímil observar um Salmãou ou um Robalo. Portanto:

Naive Bayes

- Probabilidade *a priori*
  - Reflete o quão **verossímil** é observar uma das duas espécies de peixe.
  - Se a quantidade de Salmão é **igual** à quantidade de Robalo, então é igualmente verossímil observar um Salmãou ou um Robalo. Portanto:
  - $P(\omega_1)$: probabilidade *a priori* de se observar um Robalo.

Naive Bayes

- Probabilidade *a priori*
  - Reflete o quão **verossímil** é observar uma das duas espécies de peixe.
  - Se a quantidade de Salmão é **igual** à quantidade de Robalo, então é igualmente verossímil observar um Salmãou ou um Robalo. Portanto:
  - $P(\omega_1)$: probabilidade *a priori* de se observar um Robalo.
  - $P(\omega_2)$: probabilidade *a priori* de se observar um Salmão.

Estácio

Naive Bayes

- Regra de Decisão

Naive Bayes

- Regra de Decisão
  - Informação disponível: probabilidades *a priori*.

Naive Bayes

- Regra de Decisão
  - Informação disponível: probabilidades *a priori*.
  - $\text{Decisão} = \begin{cases} \omega_1, & \text{se} \quad P(\omega_1) > P(\omega_2) \\ \omega_2, & \text{senão} \end{cases}$

Naive Bayes

- Regra de Decisão
  - Informação disponível: probabilidades *a priori*.
  - Decisão $= \begin{cases} \omega_1, & \text{se} \quad P(\omega_1) > P(\omega_2) \\ \omega_2, & \text{senão} \end{cases}$
  - Se $P(\omega_1) \gg P(\omega_2)$, a decisão a favor de ω_1 estará correta a maior parte do tempo.

Naive Bayes

- Regra de Decisão
  - Informação disponível: probabilidades *a priori*.
  - Decisão $= \begin{cases} \omega_1, & \text{se} \quad P(\omega_1) > P(\omega_2) \\ \omega_2, & \text{senão} \end{cases}$
  - Se $P(\omega_1) \gg P(\omega_2)$, a decisão a favor de ω_1 estará correta a maior parte do tempo.
  - Se $P(\omega_1) = P(\omega_2)$, essa decisão tem apenas 50% de chance de estar correta.

Estácio

Naive Bayes

- Probabilidade Condicional

Naive Bayes

- Probabilidade Condicional
  - Suponha conhecidas

Naive Bayes

- Probabilidade Condicional
  - Suponha conhecidas
    - As probabilidades *a priori* $P(\omega_j)$

Naive Bayes

- Probabilidade Condicional
  - Suponha conhecidas
    - As probabilidades *a priori* $P(\omega_j)$
    - As densidades condicionais $p(x|\omega_j), j = 1, 2$

Estácio

Naive Bayes

- Probabilidade Condicional
  - Suponha conhecidas
    - As probabilidades *a priori* $P(\omega_j)$
    - As densidades condicionais $p(x|\omega_j), j = 1, 2$
  - Suponha que o valor observador do brilho foi x.

Naive Bayes

- Probabilidade Condicional
  - Suponha conhecidas
    - As probabilidades *a priori* $P(\omega_j)$
    - As densidades condicionais $p(x|\omega_j), j = 1, 2$
  - Suponha que o valor observador do brilho foi x.
  - Como isso deve influenciar a nossa decisão em relação a que classe o peixe pertence?

Naive Bayes

- Probabilidade Condicional
  - Suponha conhecidas
    - As probabilidades *a priori* $P(\omega_j)$
    - As densidades condicionais $p(x|\omega_j), j = 1, 2$
  - Suponha que o valor observador do brilho foi x.
  - Como isso deve influenciar a nossa decisão em relação a que classe o peixe pertence?
    - Densidade de probabilidade conjunta: $p(\omega_j, x)$

Naive Bayes

- Teorema de Bayes

$$p(\omega_j, x) = P(\omega_j|x)p(x) = p(x|\omega_j)P(\omega_j)$$
$$\vdots$$
$$P(\omega_j|x) = \frac{p(x|\omega_j)P(\omega_j)}{p(x)}$$

Estácio

Naive Bayes

- Teorema de Bayes
  $$p(\omega_j, x) = P(\omega_j|x)p(x) = p(x|\omega_j)P(\omega_j)$$
  $$\vdots$$
  $$P(\omega_j|x) = \frac{p(x|\omega_j)P(\omega_j)}{p(x)}$$
- Em palavras: *posteriori* $= \dfrac{\text{verossimilhança} \times \textit{priori}}{\text{evidência}}$

Naive Bayes

- *Posteriori*

Naive Bayes

- *Posteriori*
  - Observando-se x pode-se passar da probabilidade *a priori* $P(\omega_j)$ para a probabilidade *a posteriori* $P(\omega_j|x)$.

Naive Bayes

- *Posteriori*
  - Observando-se x pode-se passar da probabilidade *a priori* $P(\omega_j)$ para a probabilidade *a posteriori* $P(\omega_j|x)$.
  - $P(\omega_j|x)$: probabilidade da classe ser ω_j dado que observou-se x.

Naive Bayes

- *Posteriori*
  - Observando-se x pode-se passar da probabilidade *a priori* $P(\omega_j)$ para a probabilidade *a posteriori* $P(\omega_j|x)$.
  - $P(\omega_j|x)$: probabilidade da classe ser ω_j dado que observou-se x.
- Verossimilhança

Naive Bayes

- *Posteriori*
  - Observando-se x pode-se passar da probabilidade *a priori* $P(\omega_j)$ para a probabilidade *a posteriori* $P(\omega_j|x)$.
  - $P(\omega_j|x)$: probabilidade da classe ser ω_j dado que observou-se x.
- Verossimilhança
  - Indica que a classe ω_j para o qual $p(x|\omega_j)$ é maior, é mais verossímil ser a verdadeira classe.

Naive Bayes

- *Posteriori*
  - Observando-se x pode-se passar da probabilidade *a priori* $P(\omega_j)$ para a probabilidade *a posteriori* $P(\omega_j|x)$.
  - $P(\omega_j|x)$: probabilidade da classe ser ω_j dado que observou-se x.
- Verossimilhança
  - Indica que a classe ω_j para o qual $p(x|\omega_j)$ é maior, é mais verossímil ser a verdadeira classe.
- Evidência

Naive Bayes

- *Posteriori*
  - Observando-se x pode-se passar da probabilidade *a priori* $P(\omega_j)$ para a probabilidade *a posteriori* $P(\omega_j|x)$.
  - $P(\omega_j|x)$: probabilidade da classe ser ω_j dado que observou-se x.
- Verossimilhança
  - Indica que a classe ω_j para o qual $p(x|\omega_j)$ é maior, é mais verossímil ser a verdadeira classe.
- Evidência
  - Fator de escala que garante que a soma das probabilidades *a posteriori* é igual a 1.

Estácio

Naive Bayes

- Classificador
  - A partir dos cálculos probabilísticos uma regra de decisão tem de ser escolhida.

Naive Bayes

- Classificador
  - A partir dos cálculos probabilísticos uma regra de decisão tem de ser escolhida.
  - Uma bastante comum é escolher a hipótese mais provável, de forma a minimizar a probabilidade de erro de classificação.

Naive Bayes

- Classificador
  - A partir dos cálculos probabilísticos uma regra de decisão tem de ser escolhida.
  - Uma bastante comum é escolher a hipótese mais provável, de forma a minimizar a probabilidade de erro de classificação.
  - Essa regra é conhecida como *máximo a posteriori*.

Naive Bayes

- Classificador
  - A partir dos cálculos probabilísticos uma regra de decisão tem de ser escolhida.
  - Uma bastante comum é escolher a hipótese mais provável, de forma a minimizar a probabilidade de erro de classificação.
  - Essa regra é conhecida como *máximo a posteriori*.
  - $\hat{y} = \underset{k \in \{1,\ldots,K\}}{argmax}\ p(C_k) \prod_{i=1}^{n} p(x_i|C_k)$

Naive Bayes

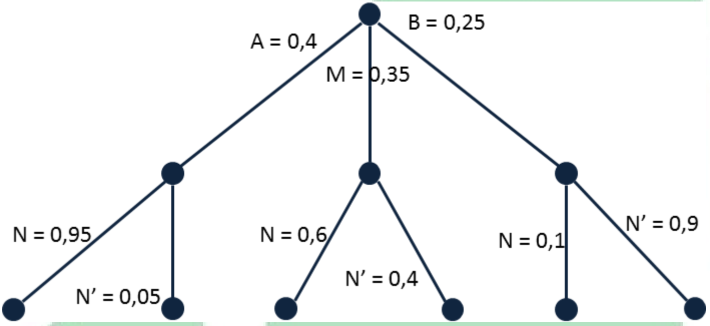
- Classificador
  - A partir dos cálculos probabilísticos uma regra de decisão tem de ser escolhida.
  - Uma bastante comum é escolher a hipótese mais provável, de forma a minimizar a probabilidade de erro de classificação.
  - Essa regra é conhecida como *máximo a posteriori*.
  - $\hat{y} = \underset{k \in \{1,\ldots,K\}}{argmax}\, p(C_k) \prod_{i=1}^{n} p(x_i|C_k)$
    - $\hat{y}$: classificação/resultado;
    - *argmax*: argumento máximo, ou seja, o maior valor;
    - K: quantidade de classes, ou seja, K classes.
    - C_k: classe k, em que $k = 1, \ldots, K$;
    - $p(C_k)$: probabilidade da classe k;
    - $p(x_i|C_k)$: probabilidade da observação x_i dada a classe k;
    - n: quantidade de observações.

Naive Bayes

- Exemplo
  - Usuários costumam avaliar *designs* preliminares de produtos. Anteriormente, 95% dos produtos de alto sucesso receberam boas notas, 60% dos produtos de sucesso moderado receberam boas notas, e 10% dos produtos de baixo sucesso receberam boas notas. Além disso, 40% dos produtos tiveram alto sucesso, 35% tiveram sucesso moderado e 25% tiveram baixo sucesso.
    (a) Qual a probabilidade de um produto ter uma boa nota?
    (b) Se um novo *desing* recebe uma boa nota, qua é a probabilidade de que ele venha a ter um alto sucesso?
    (c) Se um produto não tem uma boa nota, qual é a probabilidade de ele vir a ter um alto sucesso?

Naive Bayes

- Exemplo
  - **A** = Alto sucesso
  - **M** = Médio sucesso
  - **B** = Baixo sucesso;
  - **N** = Nota boa
  - **N'** = Nota ruim



Naive Bayes

Resolução

(a) Probabilidade de um produto ter uma boa nota

$$P(N) = P(N|A)P(A) + P(N|M)P(M) + P(N|B)P(B) = 0,615$$

(b) Probabilidade de ter um alto sucesso se receber nota boa

$$P(A|N) = \frac{P(N|A)P(A)}{P(N)} = 0,618$$

(c) Probabilidade de ter um alto sucesso se receber nota ruim

$$P(A|N') = \frac{P(N'|A)P(A)}{P(N')} \rightarrow P(N')?$$

$$P(N') = P(N'|A)P(A) + P(N'|M)P(M) + P(N'|B)P(B) = 0,385$$

$$\therefore$$

$$P(A|N') = 0,052$$

Estácio

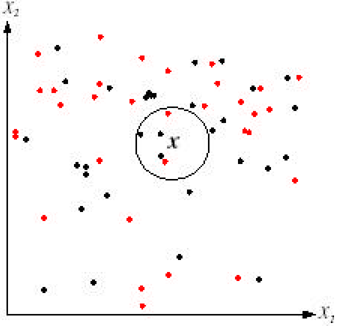
Aprendizado Supervisionado

Classificação

k-NN

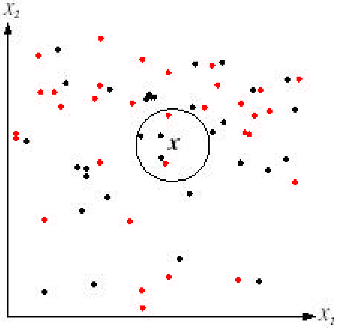
k-NN

- k-NN = k *Nearest Neighbors* ou k Vizinhos Mais Próximos.
- A estimação é baseada na probabilidade *a posteriori*.



k-NN

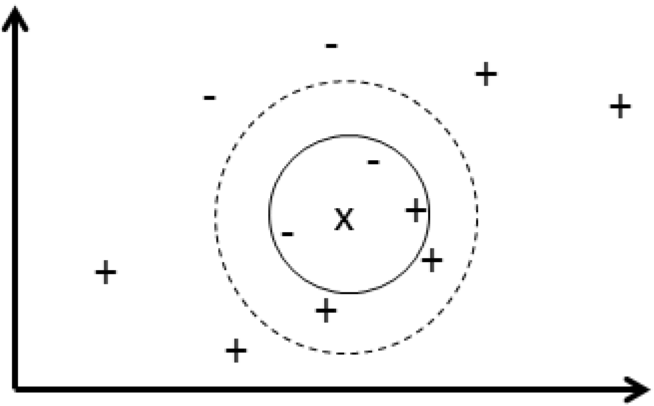
- k-NN = k *Nearest Neighbors* ou k Vizinhos Mais Próximos.
- A estimação é baseada na probabilidade *a posteriori*.



- $P(vermelho) = 2/5 \therefore P(preto) = 3/5$

k-NN

- k é um número que pode variar



Estácio

k-NN

- Como *observar* a vizinhança?

k-NN

- Como *observar* a vizinhança?
  - Distância Euclidiana (mais comum)

k-NN

- Como *observar* a vizinhança?
  - Distância Euclidiana (mais comum)
  - Distância $= \sqrt{\sum_{i=1}^{d}(p_i - q_i)^2}$

k-NN

- Como *observar* a vizinhança?
  - Distância Euclidiana (mais comum)
  - Distância $= \sqrt{\sum_{i=1}^{d}(p_i - q_i)^2}$
  - **Exemplo**: $a_1 = (1, 1)$; $a_2 = (4, 5)$
    Distância$(a_1, a_2) = \sqrt{(1-4)^2 + (1-5)^2}$
    Distância$(a_1, a_2) = \sqrt{9 + 16} = 5$

Estácio

k-NN

- Como *observar* a vizinhança?
  - Distância Euclidiana (mais comum)
  - Distância $= \sqrt{\sum_{i=1}^{d}(p_i - q_i)^2}$
  - **Exemplo**: $a_1 = (1, 1)$; $a_2 = (4, 5)$
    Distância$(a_1, a_2) = \sqrt{(1-4)^2 + (1-5)^2}$
    Distância$(a_1, a_2) = \sqrt{9 + 16} = 5$
- Depois de calcular a distância de um ponto a todos os outros, é possível saber quem são os mais próximos, e suas classes.

Estácio

Aprendizado Supervisionado

Classificação

Árvore de Decisão

Árvore de Decisão

- Segundo Mitchell [1] **Árvore de Decisão** é um método para **aproximação de funções** com **valores discretos**, onde a função aprendida é representada por uma árvore de decisão.

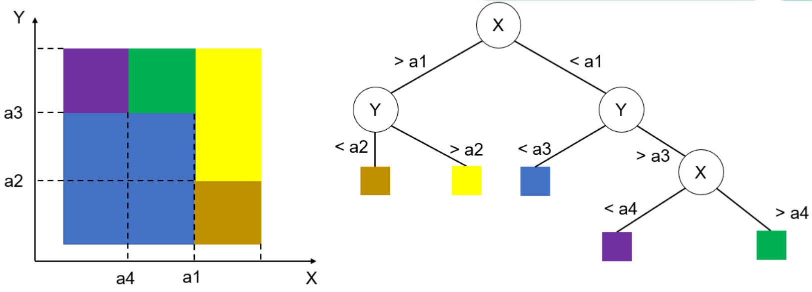
Árvore de Decisão

- Segundo Mitchell [1] **Árvore de Decisão** é um método para **aproximação de funções** com **valores discretos**, onde a função aprendida é representada por uma árvore de decisão.
- Algoritmos mais recentes permitem a criação de árvores de decisão com valores contínuos.

Estácio

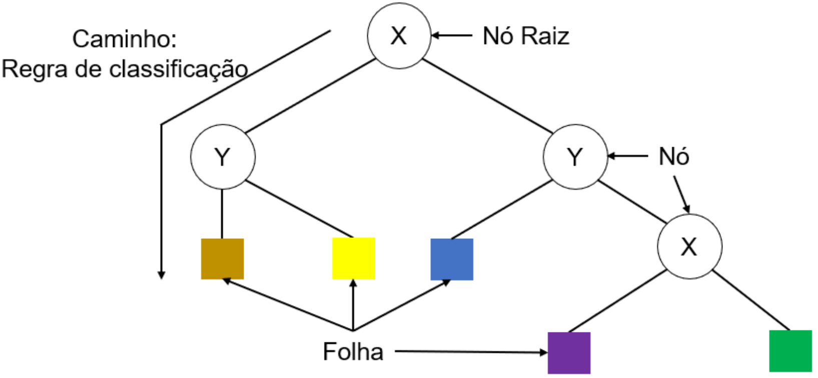
Árvore de Decisão

- Representação/Visualização de uma árvore
  - Exemplo retirado de [2]



Árvore de Decisão

- Termos técnicos de uma árvore



Árvore de Decisão

- Como construir?

Árvore de Decisão

- Como construir?
- Ideia base:

Árvore de Decisão

- Como construir?
- Ideia base:
  1. Escolher um atributo;

Árvore de Decisão

- Como construir?
- Ideia base:
  1. Escolher um atributo;
  2. Estender a árvore adicionando um ramo para cada valor do atributo;

Estácio

Árvore de Decisão

- Como construir?
- Ideia base:
  1. Escolher um atributo;
  2. Estender a árvore adicionando um ramo para cada valor do atributo;
  3. Passar os exemplos para as folhas (tendo em conta o valor do atributo escolhido);

Árvore de Decisão

- Como construir?
- Ideia base:
  1. Escolher um atributo;
  2. Estender a árvore adicionando um ramo para cada valor do atributo;
  3. Passar os exemplos para as folhas (tendo em conta o valor do atributo escolhido);
  4. Para cada folha

Árvore de Decisão

- Como construir?
- Ideia base:
  1. Escolher um atributo;
  2. Estender a árvore adicionando um ramo para cada valor do atributo;
  3. Passar os exemplos para as folhas (tendo em conta o valor do atributo escolhido);
  4. Para cada folha
     - Se todos os exemplos são da mesma classe, associar essa classe à folha;

Árvore de Decisão

- Como construir?
- Ideia base:
  1. Escolher um atributo;
  2. Estender a árvore adicionando um ramo para cada valor do atributo;
  3. Passar os exemplos para as folhas (tendo em conta o valor do atributo escolhido);
  4. Para cada folha
     - Se todos os exemplos são da mesma classe, associar essa classe à folha;
     - Senão, repetir os passos 1 a 4.

Estácio

Árvore de Decisão

- Como construir?

- Como escolher o melhor atributo?

Árvore de Decisão

- Como escolher o melhor atributo?
  - Um atributo deve ser o mais discriminante possível!

Árvore de Decisão

- Como escolher o melhor atributo?
  - Um atributo deve ser o mais discriminante possível!
  - Uma divisão, a partir de um atributo, que mantem as proporções de classes nas folhas é inútil.

Árvore de Decisão

- Como escolher o melhor atributo?
  - Um atributo deve ser o mais discriminante possível!
  - Uma divisão, a partir de um atributo, que mantem as proporções de classes nas folhas é inútil.
  - Já uma divisão que tem como resultado todos os exemplos de uma folha sendo da mesma classe, tem utilidade máxima.

Estácio

Árvore de Decisão

- Algoritmo ID3 (*Inductive Decision Tree* [3])
  - Para escolher o melhor atributo, é feito um cálculo estatístico conhecido como **ganho de informação**.

Árvore de Decisão

- Algoritmo ID3 (*Inductive Decision Tree* [3])
  - Para escolher o melhor atributo, é feito um cálculo estatístico conhecido como **ganho de informação**.
  - $G(S, A) \equiv Entropia(S) - \sum_{v \in Valores(A)} \frac{|S_v|}{|S|} Entropia(S_v)$

Estácio

Árvore de Decisão

- Algoritmo ID3 (*Inductive Decision Tree* [3])
  - Para escolher o melhor atributo, é feito um cálculo estatístico conhecido como **ganho de informação**.
  - $G(S, A) \equiv Entropia(S) - \sum_{v \in Valores(A)} \frac{|S_v|}{|S|} Entropia(S_v)$
    - Conjunto de todos os possíveis valores para o atributo A;

Árvore de Decisão

- Algoritmo ID3 (*Inductive Decision Tree* [3])
  - Para escolher o melhor atributo, é feito um cálculo estatístico conhecido como **ganho de informação**.
  - $G(S, A) \equiv Entropia(S) - \sum_{v \in Valores(A)} \frac{|S_v|}{|S|} Entropia(S_v)$
    - Conjunto de todos os possíveis valores para o atributo A;
    - Subconjunto de *S* para o qual o atributo A tem valor *v*;

Árvore de Decisão

- Algoritmo ID3 (*Inductive Decision Tree* [3])
  - Para escolher o melhor atributo, é feito um cálculo estatístico conhecido como **ganho de informação**.
  - $G(S, A) \equiv Entropia(S) - \sum_{v \in Valores(A)} \frac{|S_v|}{|S|} Entropia(S_v)$
  - Para entender essa fórmula, vamos ver seus elementos.

Árvore de Decisão

- Algoritmo ID3 (*Inductive Decision Tree* [3])
  - Para escolher o melhor atributo, é feito um cálculo estatístico conhecido como **ganho de informação**.
  - $G(S, A) \equiv Entropia(S) - \sum_{v \in Valores(A)} \frac{|S_v|}{|S|} Entropia(S_v)$
  - Para entender essa fórmula, vamos ver seus elementos.
  - Começando pela Entropia.

Estácio

Árvore de Decisão

- A Entropia é uma medida que caracteriza a pureza ou impureza de um conjunto arbitrário de exemplos [1].

Árvore de Decisão

- A Entropia é uma medida que caracteriza a pureza ou impureza de um conjunto arbitrário de exemplos [1].
- Seja um conjunto S contendo duas classes, uma positiva ($p_{\oplus} =$ proporção de exemplos positivos) e uma negativa ($p_{\ominus} =$ proporção de exemplos negativos).

Estácio

Árvore de Decisão

- A Entropia é uma medida que caracteriza a pureza ou impureza de um conjunto arbitrário de exemplos [1].
- Seja um conjunto S contendo duas classes, uma positiva ($p_{\oplus} =$ proporção de exemplos positivos) e uma negativa ($p_{\ominus} =$ proporção de exemplos negativos).
- A entropia de S é:
  $Entropia(S) \equiv -p_{\oplus} log_2 p_{\oplus} - p_{\ominus} log_2 p_{\ominus}$
  ou
  $Entropia(S) \equiv \sum_{i=1}^{c} -p_i log_2 p_i$

Estácio

Árvore de Decisão

Exemplo

S = 14 exemplos ... [9+, 5-]

Árvore de Decisão

Exemplo

S = 14 exemplos ... [9+, 5-]
Entropia(S) = Entropia([9+, 5-])

Estácio

Árvore de Decisão

Exemplo

S = 14 exemplos ... [9+, 5-]
Entropia(S) = Entropia([9+, 5-])
= $-(9/14)log_2(9/14) - (5/14)log_2(5/14) = 0,940$

Árvore de Decisão

- Exemplo ilustrativo

Dia	Tempo	Temperatura	Umidade	Vento	Jogar
D1	Ensolarado	Quente	Alta	Fraco	Não
D2	Ensolarado	Quente	Alta	Forte	Não
D3	Nublado	Quente	Alta	Fraco	Sim
D4	Chuvoso	Média	Alta	Fraco	Sim
D5	Chuvoso	Frio	Normal	Fraco	Sim
D6	Chuvoso	Frio	Normal	Forte	Não
D7	Nublado	Frio	Normal	Forte	Sim
D8	Ensolarado	Média	Alta	Fraco	Não
D9	Ensolarado	Frio	Normal	Fraco	Sim
D10	Chuvoso	Média	Normal	Fraco	Sim
D11	Ensolarado	Média	Normal	Forte	Sim
D12	Nublado	Média	Alta	Forte	Sim
D13	Nublado	Quente	Normal	Fraco	Sim
D14	Chuvoso	Média	Alta	Forte	Não

Tabela: Exemplos de treino para a decisão de jogar

Estácio

Árvore de Decisão

- Exemplo ilustrativo

Dia	Tempo	Temperatura	Umidade	Vento	Jogar
D1	Ensolarado	Quente	Alta	Fraco	Não
D2	Ensolarado	Quente	Alta	Forte	Não
D3	Nublado	Quente	Alta	Fraco	Sim
D4	Chuvoso	Média	Alta	Fraco	Sim
D5	Chuvoso	Frio	Normal	Fraco	Sim
D6	Chuvoso	Frio	Normal	Forte	Não
D7	Nublado	Frio	Normal	Forte	Sim
D8	Ensolarado	Média	Alta	Fraco	Não
D9	Ensolarado	Frio	Normal	Fraco	Sim
D10	Chuvoso	Média	Normal	Fraco	Sim
D11	Ensolarado	Média	Normal	Forte	Sim
D12	Nublado	Média	Alta	Forte	Sim
D13	Nublado	Quente	Normal	Fraco	Sim
D14	Chuvoso	Média	Alta	Forte	Não

Tabela: Exemplos de treino para a decisão de jogar

Árvore de Decisão

- Exemplo ilustrativo

Dia	Tempo	Temperatura	Umidade	Vento	Jogar
D1	Ensolarado	Quente	Alta	Fraco	Não
D2	Ensolarado	Quente	Alta	Forte	Não
D3	Nublado	Quente	Alta	Fraco	Sim
D4	Chuvoso	Média	Alta	Fraco	Sim
D5	Chuvoso	Frio	Normal	Fraco	Sim
D6	Chuvoso	Frio	Normal	Forte	Não
D7	Nublado	Frio	Normal	Forte	Sim
D8	Ensolarado	Média	Alta	Fraco	Não
D9	Ensolarado	Frio	Normal	Fraco	Sim
D10	Chuvoso	Média	Normal	Fraco	Sim
D11	Ensolarado	Média	Normal	Forte	Sim
D12	Nublado	Média	Alta	Forte	Sim
D13	Nublado	Quente	Normal	Fraco	Sim
D14	Chuvoso	Média	Alta	Forte	Não

Tabela: Exemplos de treino para a decisão de jogar

Árvore de Decisão

Ganho de Informação

$S = [9+, 5-] \rightarrow Entropia = 0,940$

Árvore de Decisão

Ganho de Informação

$S = [9+, 5-] \rightarrow Entropia = 0,940$
Atributo: Tempo = [Ensolarado, Nublado, Chuvoso]

Estácio

Árvore de Decisão

Ganho de Informação

$S = [9+, 5-] \rightarrow Entropia = 0,940$
Atributo: Tempo = [Ensolarado, Nublado, Chuvoso]
$S_{Ensolarado} = [2+, 3-];$

Árvore de Decisão

Ganho de Informação

$S = [9+, 5-] \rightarrow Entropia = 0,940$
Atributo: Tempo = [Ensolarado, Nublado, Chuvoso]
$S_{Ensolarado} = [2+, 3-];$
$S_{Nublado} = [4+, 0-];$

Estácio

Árvore de Decisão

Ganho de Informação

$S = [9+, 5-] \rightarrow Entropia = 0,940$

Atributo: Tempo = [Ensolarado, Nublado, Chuvoso]

$S_{Ensolarado} = [2+, 3-];$

$S_{Nublado} = [4+, 0-];$

$S_{Chuvoso} = [3+, 2-];$

Árvore de Decisão

Ganho de Informação

$S = [9+, 5-] \rightarrow Entropia = 0,940$

Atributo: Tempo = [Ensolarado, Nublado, Chuvoso]

$S_{Ensolarado} = [2+, 3-]$;

$S_{Nublado} = [4+, 0-]$;

$S_{Chuvoso} = [3+, 2-]$;

$$G(S, A) \equiv Entropia(S) - \sum_{v \in Valores(A)} \frac{|S_v|}{|S|} Entropia(S_v)$$

Estácio

Árvore de Decisão

Ganho de Informação

$S = [9+, 5-] \rightarrow Entropia = 0,940$
Atributo: Tempo = [Ensolarado, Nublado, Chuvoso]
$S_{Ensolarado} = [2+, 3-];$
$S_{Nublado} = [4+, 0-];$
$S_{Chuvoso} = [3+, 2-];$

$$G(S, A) \equiv Entropia(S) - \sum_{v \in Valores(A)} \frac{|S_v|}{|S|} Entropia(S_v)$$

$$G(S, Tempo) \equiv 0,940 - (5/14) Entropia(S_{Ensolarado}) - (4/14) Entropia(S_{Nublado}) - (5/14) Entropia(S_{Chuvoso})$$

Estácio

Árvore de Decisão

Ganho de Informação

$S = [9+, 5-] \rightarrow Entropia = 0,940$
Atributo: Tempo = [Ensolarado, Nublado, Chuvoso]
$S_{Ensolarado} = [2+, 3-];$
$S_{Nublado} = [4+, 0-];$
$S_{Chuvoso} = [3+, 2-];$

$$G(S, A) \equiv Entropia(S) - \sum_{v \in Valores(A)} \frac{|S_v|}{|S|} Entropia(S_v)$$

$$G(S, Tempo) \equiv 0,940 - (5/14) Entropia(S_{Ensolarado}) - (4/14) Entropia(S_{Nublado}) - (5/14) Entropia(S_{Chuvoso})$$

$$G(S, Tempo) \equiv 0,940 - (5/14)0,971 - (4/14)0 - (5/14)0,971$$

Árvore de Decisão

Ganho de Informação

$S = [9+, 5-] \rightarrow Entropia = 0,940$
Atributo: Tempo = [Ensolarado, Nublado, Chuvoso]
$S_{Ensolarado} = [2+, 3-];$
$S_{Nublado} = [4+, 0-];$
$S_{Chuvoso} = [3+, 2-];$

$$G(S, A) \equiv Entropia(S) - \sum_{v \in Valores(A)} \frac{|S_v|}{|S|} Entropia(S_v)$$

$$G(S, Tempo) \equiv 0,940 - (5/14)Entropia(S_{Ensolarado}) \\ -(4/14)Entropia(S_{Nublado}) - (5/14)Entropia(S_{Chuvoso})$$

$$G(S, Tempo) \equiv 0,940 - (5/14)0,971 - (4/14)0 - (5/14)0,971$$
$$G(S, Tempo) \equiv 0,940 - 0,347 - 0 - 0,347$$

Estácio

Árvore de Decisão

Ganho de Informação

$S = [9+, 5-] \rightarrow Entropia = 0,940$
Atributo: Tempo = [Ensolarado, Nublado, Chuvoso]
$S_{Ensolarado} = [2+, 3-]$;
$S_{Nublado} = [4+, 0-]$;
$S_{Chuvoso} = [3+, 2-]$;

$$G(S, A) \equiv Entropia(S) - \sum_{v \in Valores(A)} \frac{|S_v|}{|S|} Entropia(S_v)$$

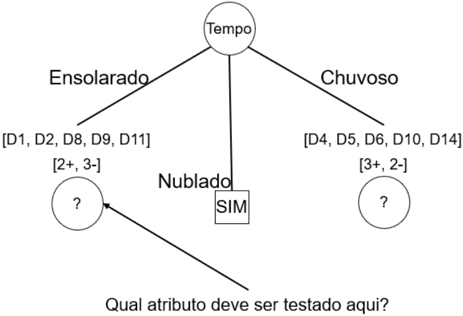
$$G(S, Tempo) \equiv 0,940 - (5/14)Entropia(S_{Ensolarado}) \\ -(4/14)Entropia(S_{Nublado}) - (5/14)Entropia(S_{Chuvoso})$$

$$G(S, Tempo) \equiv 0,940 - (5/14)0,971 - (4/14)0 - (5/14)0,971$$
$$G(S, Tempo) \equiv 0,940 - 0,347 - 0 - 0,347$$
$$G(S, Tempo) \equiv 0,246$$

Árvore de Decisão

Ganho de Informação

$Ganho(S, Tempo) = 0,246$
$Ganho(S, Temperatura) = 0,029$
$Ganho(S, Umidade) = 0,151$
$Ganho(S, Vento) = 0,048$



Árvore de Decisão

Próximo Atributo

$S_{Ensolarado} = [\text{D1, D2, D8, D9, D11}]$

$$Ganho(S_{Ensolarado}, Umidade) = 0,970 - (3/5)0 - (2/5)0 = 0,970$$

$$Ganho(S_{Ensolarado}, Temperatura) = 0,970 - (2/5)0 - (2/5)1 - (1/5)0 = 0,570$$

$$Ganho(S_{Ensolarado}, Vento) = 0,970 - (2/5)1 - (3/5)0,918 = 0,019$$

Referências I

[1] Tom M. Mitchell. **Machine Learning**. ISBN: 0070428077. McGraw-Hill Science/Engineering/-Math, mar. de 1997.

[2] Teresa B. Ludermir. **Ávores de Decisão: Sistemas Inteligentes**. Aula ministrada para o curso de Pós-Graduação em Ciência da Computação, CIn-UFPE. 2012.

[3] J. R. Quinlan. "Induction of decision trees". Em: **Machine Learning** 1.1 (1986), pp. 81–106.