Inteligência Artificial

Aula 27- Aprendizagem de Máquina: Classificação [1]

Sílvia M.W. Moraes

[1] Este material não pode ser reproduzido ou utilizado de forma parcial sem a permissão dos autores.

Sinopse

- Nesta aula, continuamos a falar sobre **aprendizagem de máquina.**
- Este material foi construído com base no material sobre Data Mining dos professores Eamonn Keogh(University of California), Rodrigos Barros, Duncan e Renata de Paris e também nos capítulos:
  - 4 - Inteligência Artificial: Uma abordagem de Aprendizagem de Máquina: Facelli e outros.
  - 18 do livro Artificial Intelligence – a Modern Approach: Russel & Norvig

Sumário

Aulas anteriores

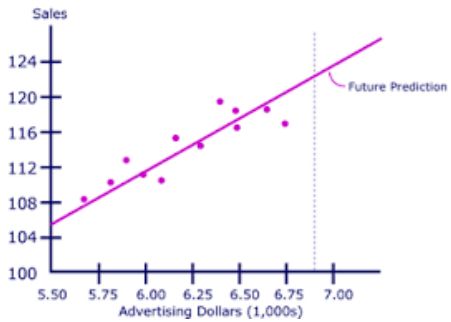
- Agente Reativos e Cognitivos
- Solução de Problemas: Algoritmos de busca
- Planejamento Clássico
- Introdução à Raciocínio Probabilistico
- Introdução à Aprendizagem de Máquina
  - Pré-processamento
  - Agrupamento

Paradigmas e Tarefas de Aprendizagem

- **Paradigma de aprendizagem** é definido pela natureza do problema. Tipo de realimentação usada pelo algoritmo para aprender.
  - Podem ser:
    - **Supervisionado**: aprendizagem de uma função h a partir de exemplos (amostras rotuladas), de entradas (x) e saídas correspondentes ($f(x)$). Com crítica referente ao erro da saída.
    - **Não-supervisionado**: aprendizagem a partir de as amostras não são rotuladas. Essa abordagem não usa os atributos de saída. Sem critica, usa regularidades e propriedades estatísticas dos dados.
    - **Por reforço**: processo de aprendizagem baseado em punição e recompensa. Reforça uma ação positiva e penaliza, uma negativa. Critica apenas de desempenho.

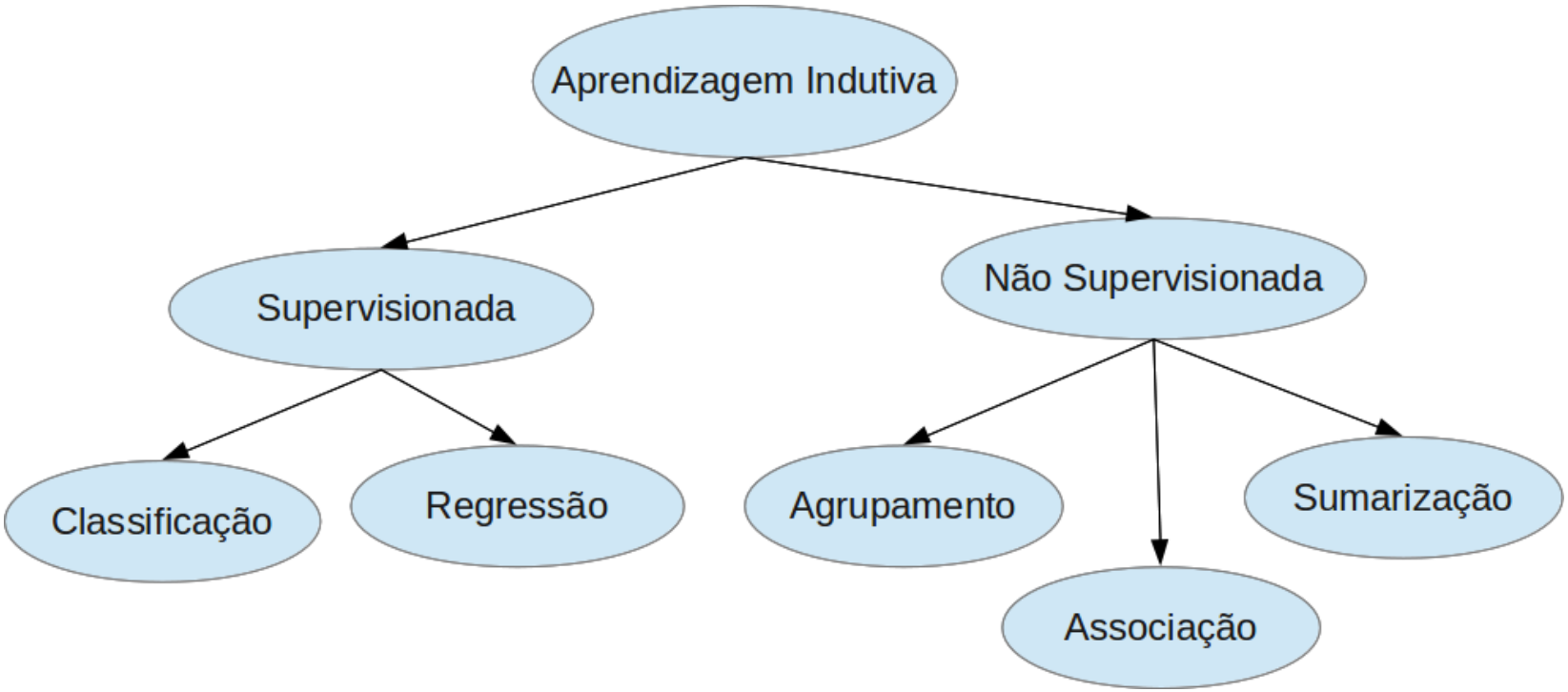
Paradigmas e Tarefas de Aprendizagem

- As **tarefas de aprendizagem** podem ser: **preditivas** ou **descritivas**
  - **preditivas:** tarefa supervisionada, sua meta é encontrar uma função (modelo ou hipótese) a partir dos dados de treino que possa ser usada para prever um rótulo (classe) ou valor de um novo exemplo.
    - Ex: **classificação** (rótulos discretos), **regressão** (rótulos contínuos)



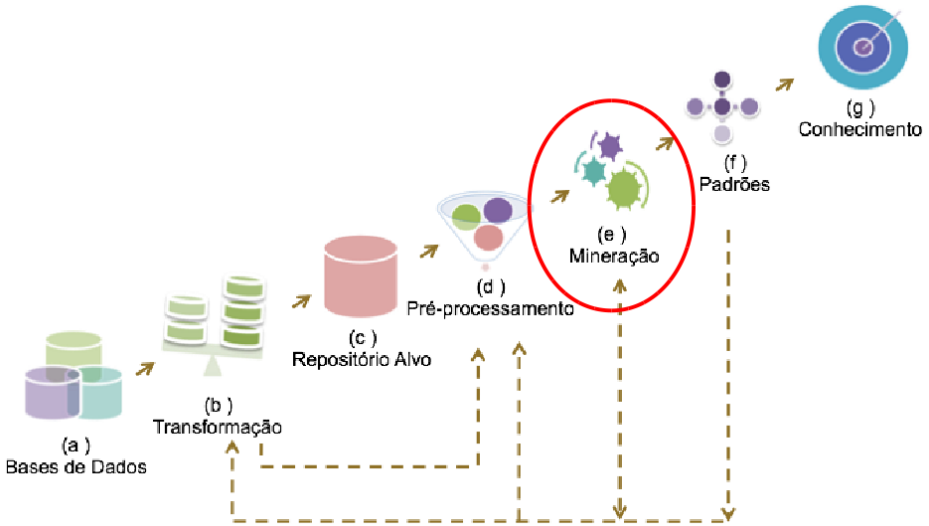
Paradigmas e Tarefas de Aprendizagem

- Resumo:



Processo de Descoberta de Conhecimento

- **Knowledge Discovery in Databases (KDD)**: consiste em uma série de passos bem definida cujo meta é transformar dados em conhecimento.
  (e) Mineração :
  - Usa Algoritmos de aprendizado de máquina
  - Análise de uma séries de dados para compreensão do domínio
  - Resultados compreensíveis e especialmente úteis



Classificação: Conceito

- **Objetivo: classificação de dados é o processo de automaticamente atribuir um (single label) ou mais rótulos (multi-label), ditos classes, aos dados.**

Classificação: Características

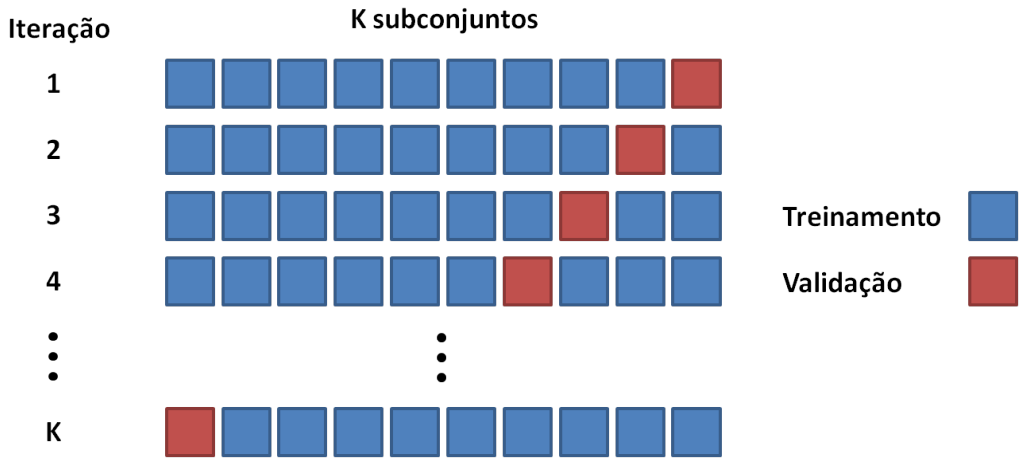
- **Características:**
  - É uma **tarefa preditiva**, supervisionada que exige que os **dados usados para definir o modelo estejam rotulados**.
  - Os rótulos (classes) são pré-definidos.
  - Nessa tarefa, **os dados são divididos inicialmente em 2 subconjuntos** disjuntos:
    - **Conjunto de treinamento:** é usado para treinar o algoritmo durante a etapa de fase aprendizagem.
      Este conjunto é subdividido em 2 novos conjuntos disjuntos:
      - Subconjunto **de estimação**: usado para selecionar o modelo;
      - Subconjunto **de validação:** usado para testar e validar o modelo.
    - **Conjunto de teste:** é usado para validar o modelo na fase de generalização (teste final).
  - No mínimo, 2 conjuntos: treinamento (~80% das amostras) e teste (~20% das amostras) .

Classificação: Características

- A **subdivisão em 3 conjuntos é interessante**, pois
  - nos permite avaliar o desempenho de vários modelos candidatos antes de escolhermos o melhor (quando usamos os dois 2 subconjuntos do treinamento).
  - nos resguarda da possibilidade do melhor modelo estar excessivamente ajustado ao subconjunto de validação (essa garantia é dada pelo conjunto de teste).

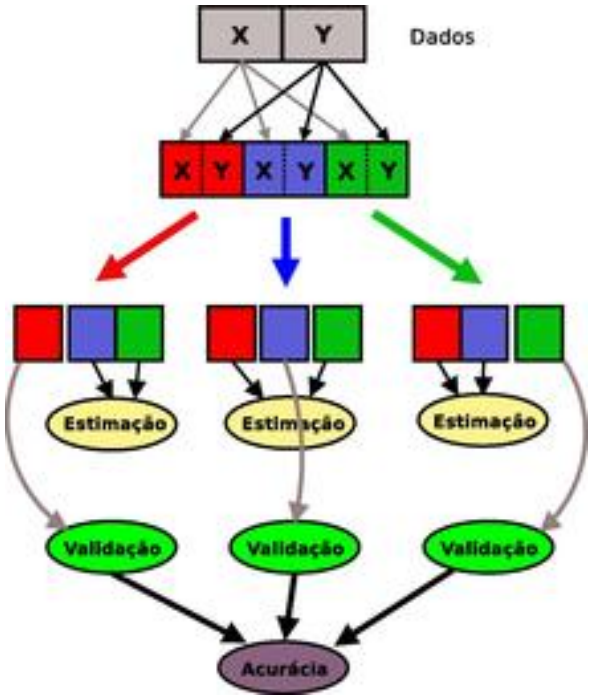
Classificação: Características

- **Validação Cruzada:**
  - O conjunto de N dados é subdividido em K subconjuntos ($K > 1$, sendo N divisivel por K).
  - O modelo é treinamento todos os subconjuntos, exceto um. O subconjunto deixado de fora é usado para testar e validar o modelo.
  - O procedimento é repetido K vezes sempre deixando um subconjunto diferente de fora.



Classificação: Características

- **Validação Cruzada:**
  - O desempenho do modelo é medido pela média do erro quadrado obtido na validação de todas essas K tentativas.
  - Mais processamento, mas é vantajoso quando o melhor conjunto de treinamento não é conhecido.



Classificação: Exemplo

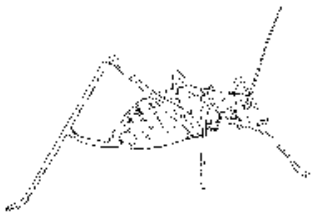
- **Exemplo:** Classificação de tipos de gafanhoto:


X


Classificação: Exemplo

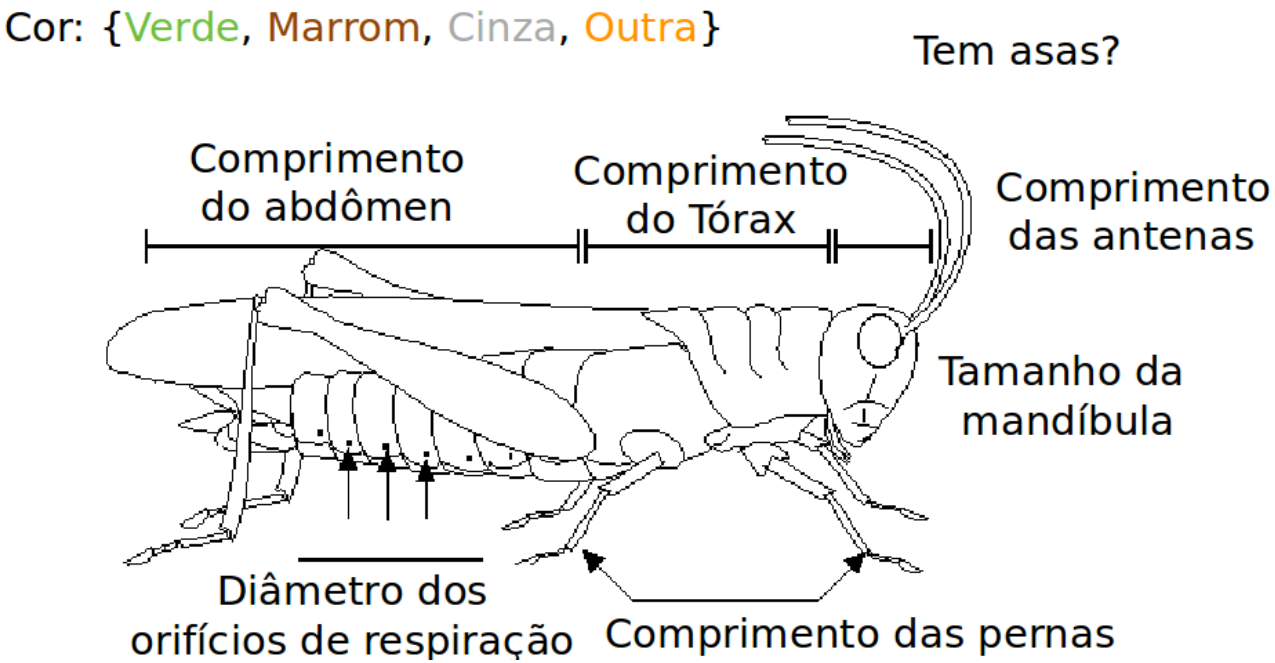
- Considerando 5 exemplos de Esperança (tipo de gafanhoto verde) e 5 do Gafanhoto.

- Qual o tipo do inseto abaixo ?



Classificação: Exemplo

- Para viabilizar a classificação, precisamos identificar um conjunto de características que nos permite distinguir as classes dos insetos, tais como:



Classificação: Exemplo

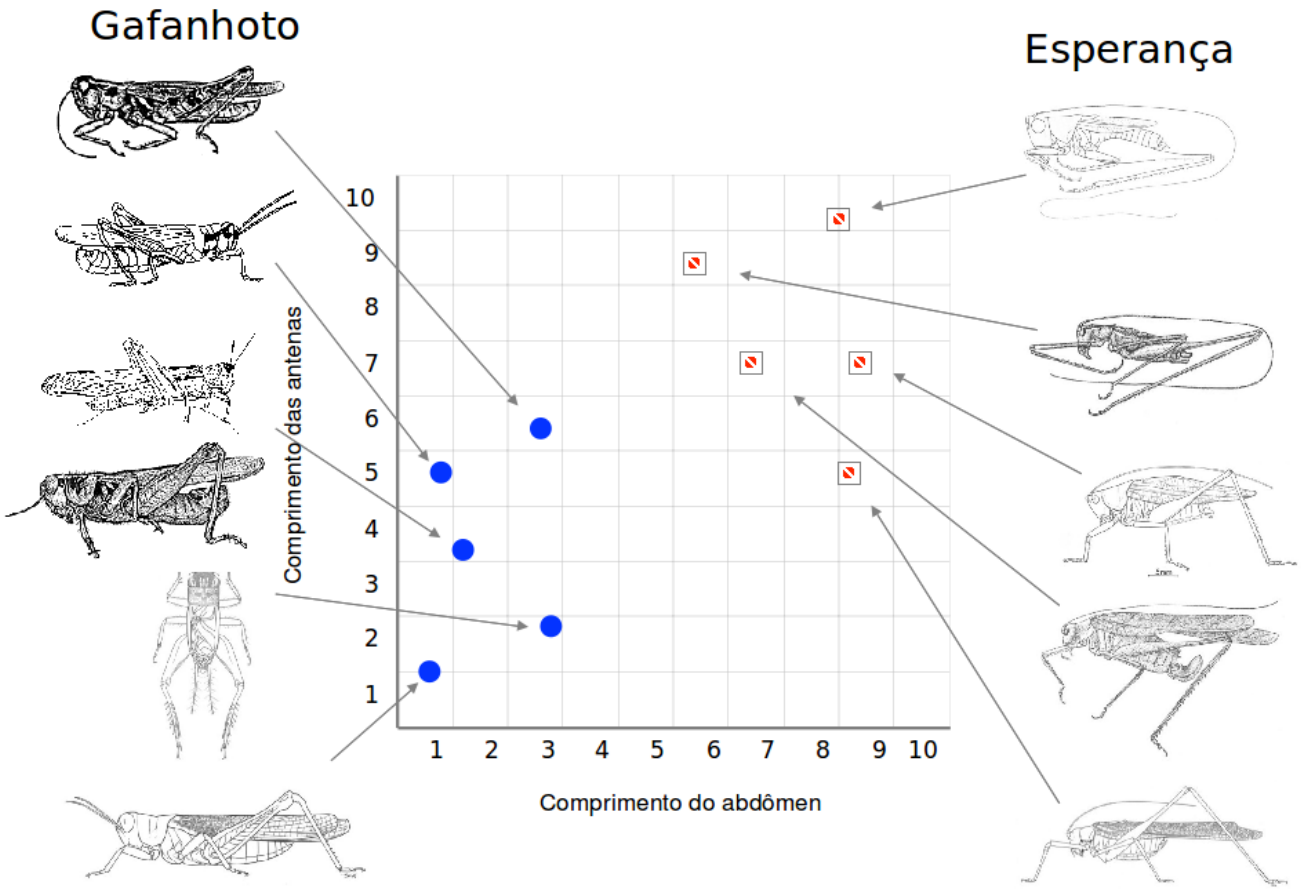
- Considerando os atributos comprimento do abdômen e das antenas, vamos tentar definir o modelo:

ID do inseto	Comp. do abdômen	Comp. das antenas	Classe do inseto
1	2.7	5.5	**Gafanhoto**
2	8.0	9.1	**Esperança**
3	0.9	4.7	**Gafanhoto**
4	1.1	3.1	**Gafanhoto**
5	5.4	8.5	**Esperança**
6	2.9	1.9	**Gafanhoto**
7	6.1	6.6	**Esperança**
8	0.5	1.0	**Gafanhoto**
9	8.3	6.6	**Esperança**
10	8.1	4.7	**Esperança**

- Qual a classe do inseto cujo abdômen mede 5.1 e as antenas

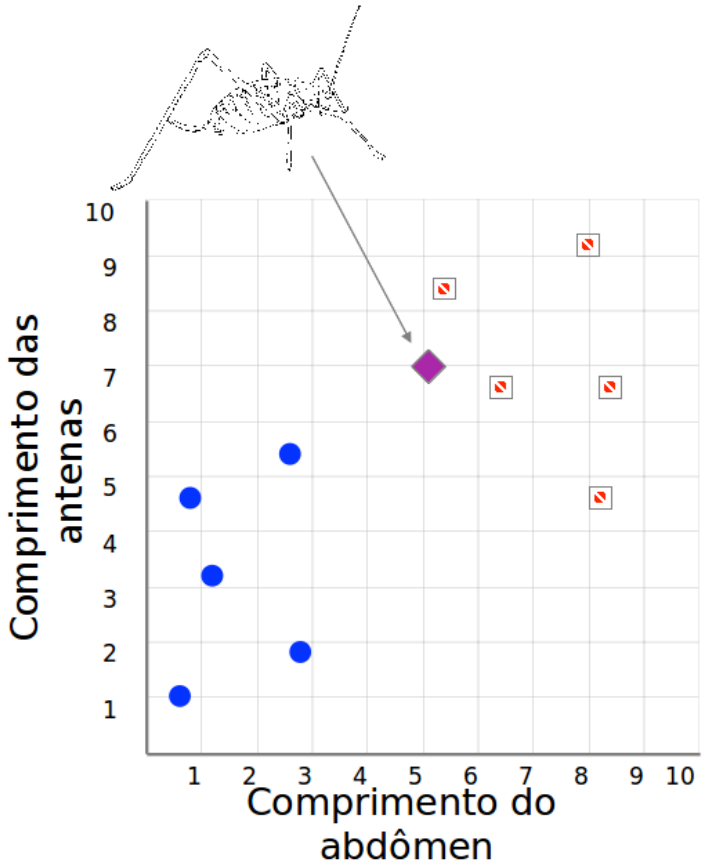
Classificação: Exemplo

- Colocando os atributos em um gráfico, conseguimos observar as classes dos gafanhotos.



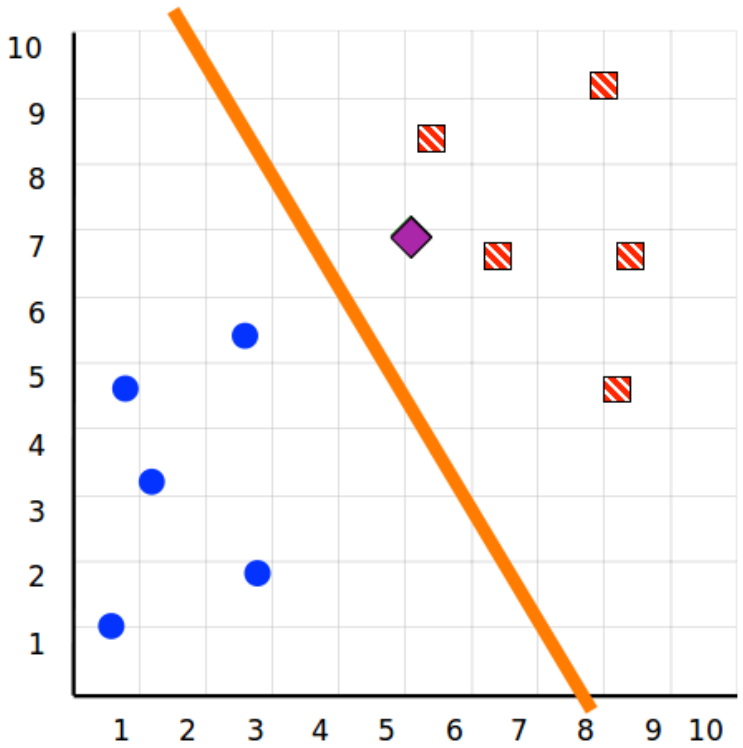
Classificação: Exemplo

- Qual a classe do inseto abaixo cujo abdômen mede 5.1 e as antenas 7.0mm?



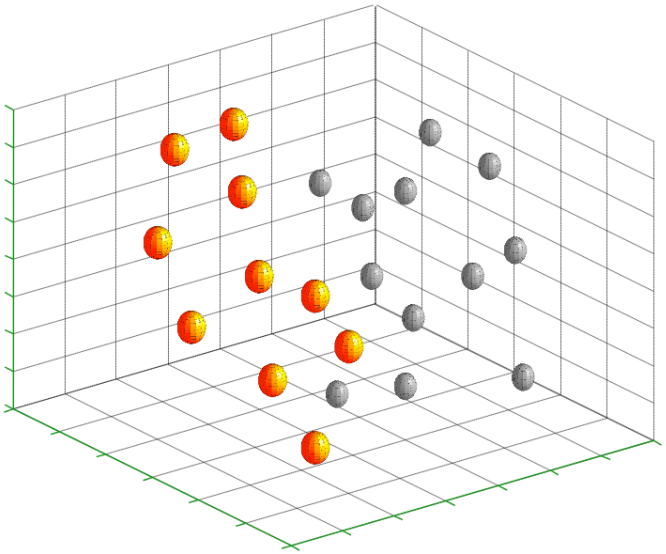
Classificação: Exemplo

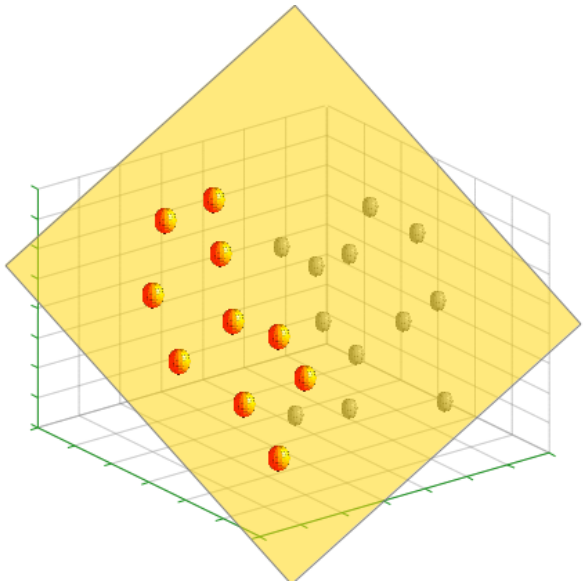
- Classificador Linear Simples
- se exemplo desconhecido está acima da linha
  então classe é Esperança
  senão classe é Gafanhoto



Classificador Linear

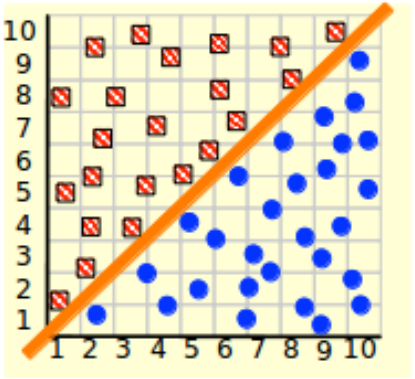
- Classificador Linear Simples:
  - Pode ser um usado em espaços dimensionais maiores
  - Nesse caso, a divisão será feita por um hiperplano.



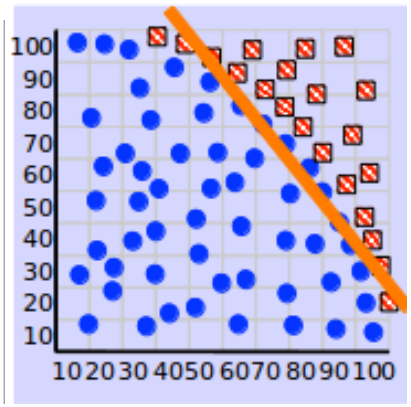


Classificador Linear

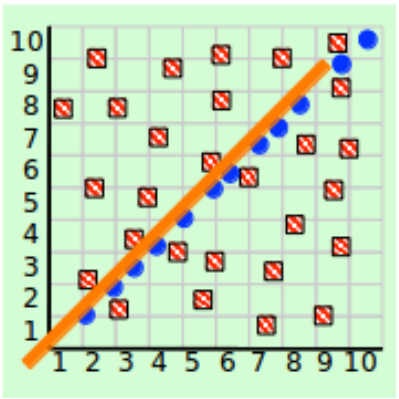
- Classificador Linear Simples:
  - Limitação: resolve apenas problemas cujas classes são linearmente separáveis.



Perfeito



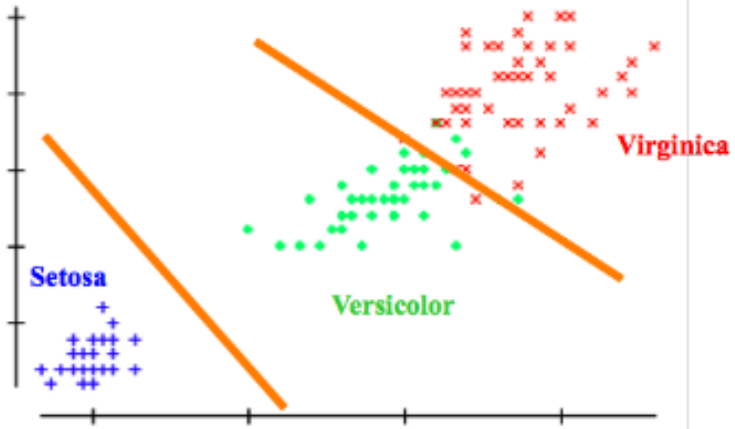
Muito Bom



Inutil

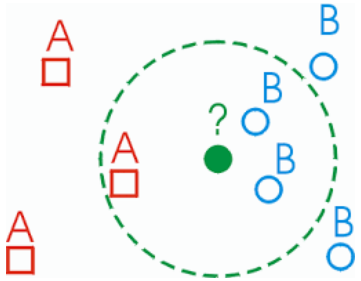
Generalização de um Classificador Linear

- **Exemplo: Base de Dados da Planta Iris**
  - 150 amostras de Iris Setosa, Virginica e Versicolor (conjunto perfeitamente balanceado)
- Podemos generalizar o **classificador linear para problemas de N classes ao ajustar $N-1$ hiperplanos**.
- Neste caso, primeiro aprendemos a linha que (perfeitamente) discrimina entre
  - Setosa e Virginica/Versicolor e,
  - depois, aprendemos a aproximadamente discriminar entre Virginica e Versicolor



Algoritmo k-NN

- Algoritmo dos k Vizinhos mais Próximos (**k**-**N**earest **N**eighbour): considera que objetos com características semelhantes pertencem ao mesmo grupo.



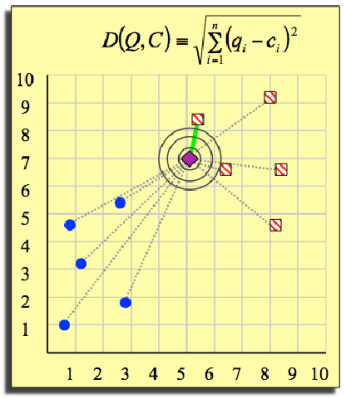
Algoritmo k-NN

- Algoritmo baseado em memória (Lazy: computação adiada para a fase de classificação)
  - **Etapa de Treinamento:**
    - Não gera modelo: apenas memorização dos dados rotulados
  - **Etapa de Generalização:**
    - Classifica uma nova amostra levando em consideração os **k** vizinhos mais próximos.
    - Usa uma medida de distância para calcular a proximidade dos dados vizinhos (distância euclidiana é bem usual).
    - Cada vizinho k próximo ao dado vota em uma classe.
    - A classe mais votada passa a ser a da nova amostra.

Algoritmo k-NN

```
Inicialização
  Preparar o conjunto de dados (treinamento e teste)
  Informar o valor de k
Para cada nova amostra do conjunto de teste faça
{
  Calcular a distância para todas as amostras do conjunto de treinamento
  Determinar o conjunto das k amostras mais próximas
  Determinar o rótulo mais representativo entre os k vizinhos
}

retornar o conjunto de teste rotulado
```



Algoritmo k-NN

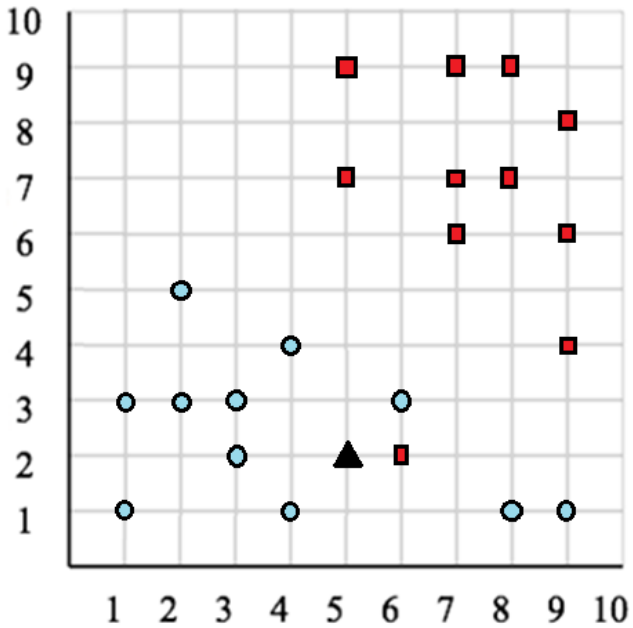
- **Aspectos Positivos**
  - Treinamento é simples (apenas armazenamento dos objetos rotulados)
  - Constrói aproximações locais da função objetivo, pois são diferentes para cada objeto novo que foi classificado
  - Aplicável mesmo em problemas complexos.
  - Incremental (quando novas amostras de treinamento estão disponíveis, basta memorizá-las)

Algoritmo k-NN

- **Aspectos Negativos**
  - Não gera um modelo
  - Dependente da medida de distância (importante normalizar os dados)
  - Predição pode ser custosa para muito objetos
  - Alta dimensionalidade dos atributos afeta negativamente os algoritmos baseados em distância (quanto maior o número de atributos mais a distância do mais próximo aproxima-se do mais distante).

Algoritmo k-NN

- **Atividade I:** Encontre a classe do objeto desconhecido (23), considerando k=3.



Objeto	X	y	Classe
1	5	9	□
2	7	9	□
3	8	9	□
4	9	8	□
5	5	7	□
6	7	7	□
7	8	7	□
8	7	6	□
9	9	6	□
10	9	4	□
11	6	2	□
12	2	5	○
13	4	4	○
14	1	3	○
15	2	3	○
16	3	3	○
17	6	3	○
18	3	2	○
19	1	1	○
20	4	1	○
21	8	1	○
22	9	1	○
23	5	2	?