

# Ciência de Dados e Aprendizado de Máquina

Profa. Leticia T. M. Zoby

[leticia.zoby@udf.edu.br](mailto:leticia.zoby@udf.edu.br)

# *Machine Learning* - Aprendizagem de Máquina

- ▶ Modelo Supervisionado

- ▶ Classificação

- ▶ Naives Bayes

- ▶ Árvore de Decisão

- ▶ Redes Neurais

- ▶ **K-vizinhos mais próximos**

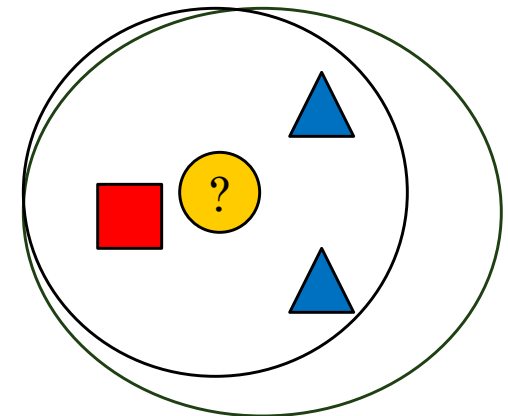
- ▶ Máquina de vetores-suporte (*Support Vector Machine* - SVM)

**Vizinho Mais Próximo –  
K-NN (*K-Nearest Neighbors*)**

# Aprendizagem de Máquina

## Vizinho Mais Próximo - K-NN (K-Nearest Neighbors)

- O K-NN considera que os registros do conjunto de dados correspondem a pontos no  $R_n$ , em que cada atributo corresponde a uma dimensão deste espaço.
- Pode ser utilizado tanto para a realização de classificação como regressão.
- No método K-NN, o conjunto de dados é armazenado. Quando um novo registro deve ser classificado, este registro é comparado a todos os registros do conjunto de treinamento para identificar k vizinhos mais próximos, de acordo com alguma métrica.



# Aprendizagem de Máquina

## Vizinho Mais Próximo - K-NN (K-Nearest Neighbors)

- Para utilizar o KNN é necessário:
  - Um conjunto de exemplos de treinamento.
  - Definir uma métrica para calcular a distância entre os exemplos de treinamento.
  - Definir o valor de K (o número de vizinhos mais próximos que serão considerados pelo algoritmo).

# Aprendizagem de Máquina

## Vizinho Mais Próximo - K-NN (K-Nearest Neighbors)

- Classificar um exemplo desconhecido com o algoritmo KNN consiste em:
  - Calcular a distância entre o exemplo desconhecido e o outros exemplos do conjunto de treinamento.
  - Identificar os K vizinhos mais próximos.
  - Utilizar o rótulo da classe dos vizinhos mais próximos para determinar o rótulo de classe do exemplo desconhecido (votação majoritária).

# Aprendizagem de Máquina

## Vizinho Mais Próximo - K-NN (K-Nearest Neighbors)

- Qualquer que seja a métrica utilizada, ela deve resultar em um valor ordinal.
- As métricas mais utilizadas no K-NN são a **Euclidiana** (a mais comum) e a de **Manhattan**.

# Aprendizagem de Máquina

## Distâncias:

- Distância Euclidiana: simplesmente é a distância geométrica no espaço multidimensional. Ela é calculada como:

$$d(\underline{\mathbf{x}}, \underline{\mathbf{y}}) = \sqrt{\sum_{i=1}^p (x_i - y_i)^2}$$

- Quadrado da dist  $d(\underline{\mathbf{x}}, \underline{\mathbf{y}}) = \sum_{i=1}^p (x_i - y_i)^2$

- Distância Manha  $d(\underline{\mathbf{x}}, \underline{\mathbf{y}}) = \sum_{i=1}^p |x_i - y_i|$



# Aprendizagem de Máquina

## Vizinho Mais Próximo - K-NN (K-Nearest Neighbors)

- Exemplo: análise de crédito

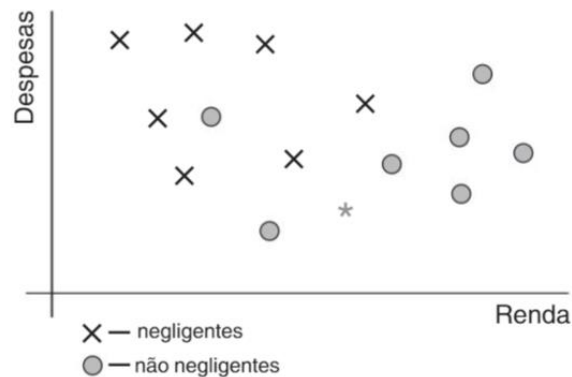


Figura 1. Conjunto contendo dados sobre clientes que receberam crédito.

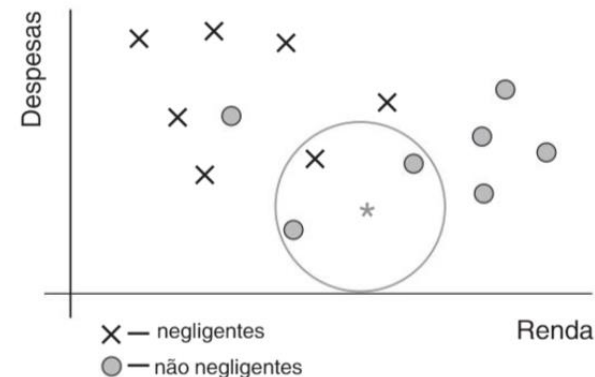
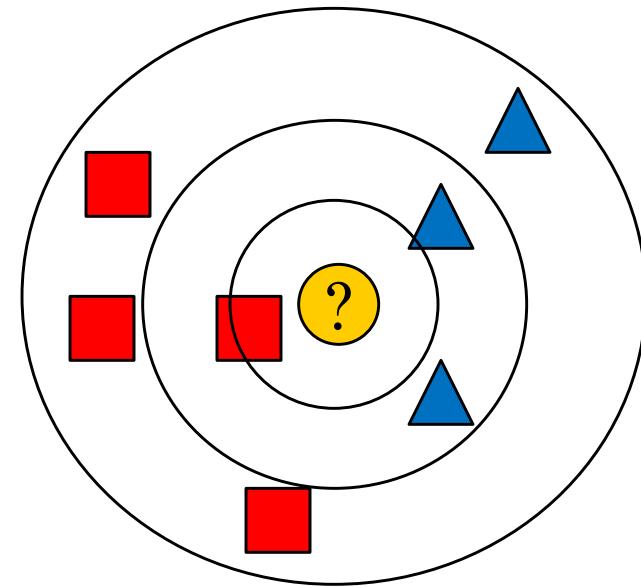


Figura 2. Seleção da vizinhança do registro "\*" durante o processamento do k-NN no exemplo apresentado (k = 3).

# Aprendizagem de Máquina

## Vizinho Mais Próximo - K-NN (K-Nearest Neighbors)

- $K = 1$ 
  - Pertence a classe de quadrados.
- $K = 3$ 
  - Pertence a classe de triângulos.
- $K = 7$ 
  - Pertence a classe de quadrados.



# Aprendizagem de Máquina

## Vizinho Mais Próximo - K-NN (K-Nearest Neighbors)

- Como escolher o valor de K?
  - Se K for muito pequeno, a classificação fica sensível a pontos de ruído.
  - Se k é muito grande, a vizinhança pode incluir elementos de outras classes.
- Além disso, é necessário sempre escolher um valor ímpar para K, assim se evita empates na votação.

# Aprendizagem de Máquina

## Vizinho Mais Próximo - K-NN (K-Nearest Neighbors)

- Pseudo-código para treinamento e teste:

---

**Algoritmo 1:** *Treinamento kNN.*

---

**Entrada:** conjunto de treinamento  $T = \{x_i, y_i\}_{i=1}^n$ , valor de  $k$  e uma medida de distância  $d(\cdot, \cdot)$

**Saída** : classificador kNN

- 1 armazenar o conjunto de treinamento e o valor de  $k$ ;
- 

---

**Algoritmo 2:** *Teste kNN.*

---

**Entrada:** classificador kNN e um objeto  $x$  cuja classe é desconhecida

**Saída** : classe  $y$  atribuída a  $x$

- 1 buscar pelos  $k$  objetos mais próximos a  $x$  no conjunto de dados de treinamento do classificador kNN informado;
  - 2 dentre os  $k$  vizinhos, determinar  $y$  como a classe mais frequente entre eles, resolvendo possíveis empates de maneira arbitrária;
- 

- Em Python:

```
from sklearn.neighbors import KNeighborsClassifier
```

# Aprendizagem de Máquina

## Vizinho Mais Próximo - K-NN (K-Nearest Neighbors)

- **Vantagens:**

- Técnica simples e facilmente implementada.
- Bastante flexível.
- Em alguns casos apresenta ótimos resultados.

- **Desvantagens:**

- Classificar um exemplo desconhecido pode ser um processo computacionalmente complexo. Requer um calculo de distancia para cada exemplo de treinamento.
  - Pode consumir muito tempo quando o conjunto de treinamento é muito grande.
- A precisão da classificação pode ser severamente degradada pela presença de ruído ou características irrelevantes.

# Máquinas de Vetores Suporte

## (SVM - *Support Vector Machines*)

# Aprendizagem de Máquina

## Máquinas de Vetores Suporte (SVM - *Support Vector Machines*)

- É um dos mais efetivos para a tarefa de Classificação
- No algoritmo, o conjunto de dados de entrada é utilizada para construir uma função de decisão  $f(x)$ , tal que:

$$\begin{aligned} \text{Se } f(x_i) \geq 0, \text{ então } y_i &= 1 \\ \text{Se } f(x_i) < 0, \text{ então } y_i &= -1 \end{aligned}$$

- Considere um conjunto de dados de treinamento da forma  $\{(x_i, y_i)\}$ , em que  $x_i$  corresponde ao vetor de atributos previsores, e  $y_i \in \{-1, 1\}$ .

# Aprendizagem de Máquina

## Máquinas de Vetores Suporte (SVM - *Support Vector Machines*)

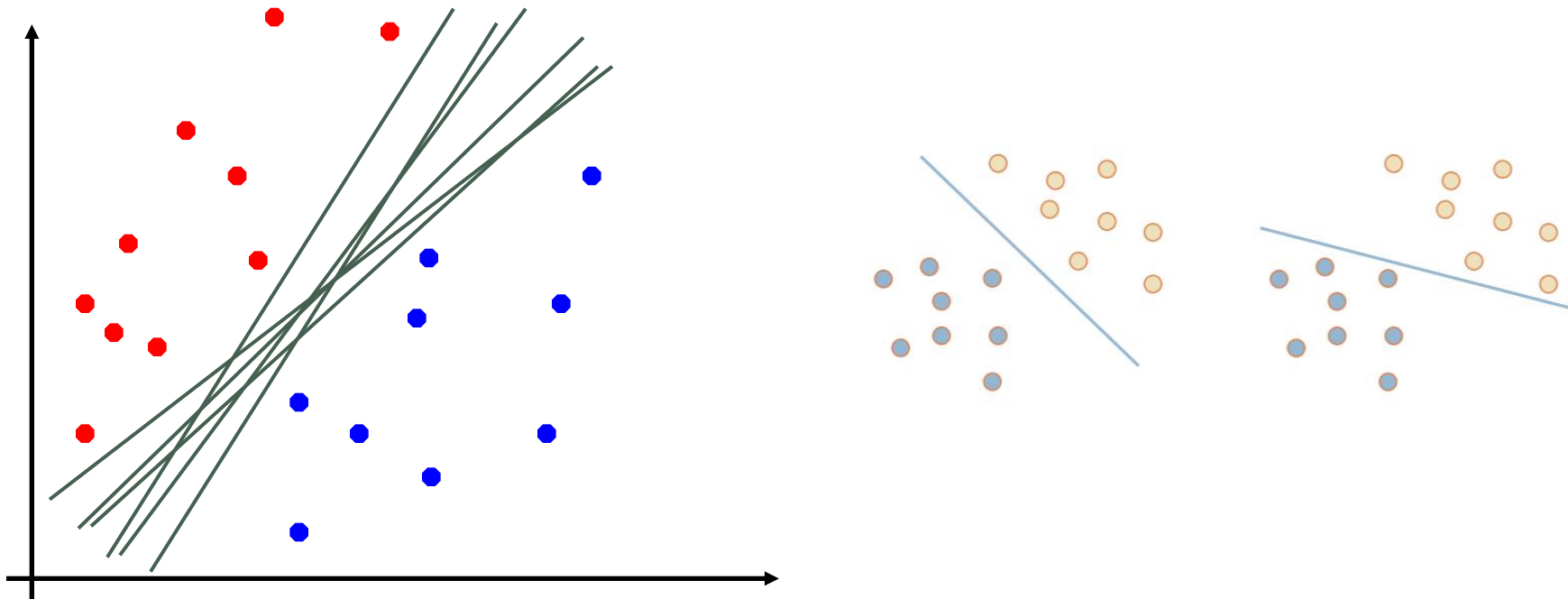
- Proposto em 1995 pelo russo **Vladimir Vapnik**.
- Consiste em um método de aprendizado que tenta encontrar a **maior margem** para separar diferentes classes de dados.
- A essência do SVM é a construção de um **hiperplano ótimo**, de modo que ele possa separar diferentes classes de dados com a maior margem possível.



# Aprendizagem de Máquina

## Máquinas de Vetores Suporte

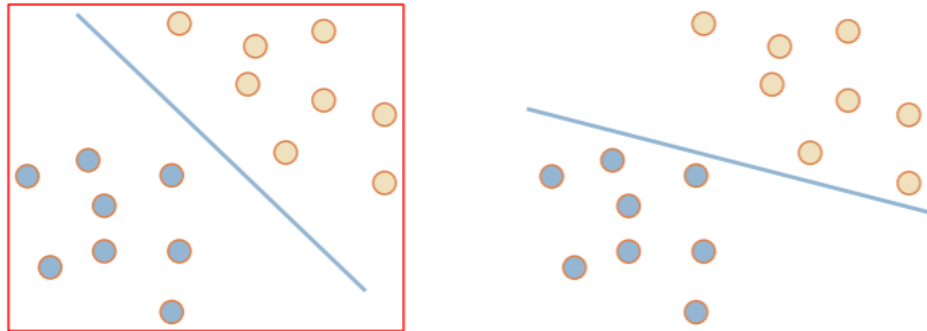
- Resultados comparáveis aos obtidos por outros algoritmos de aprendizado, como RNA (Redes Neurais Artificiais).
- Ideia geral: Perceptron é capaz de construir uma fronteira se os dados forem linearmente separáveis. Mas qual fronteira é a melhor?



# Aprendizagem de Máquina

## Máquinas de Vetores Suporte

- Ideia geral:
  - SVM trabalha com a maximização da margem
  - A fronteira mais distante dos dados de treinamento é a melhor



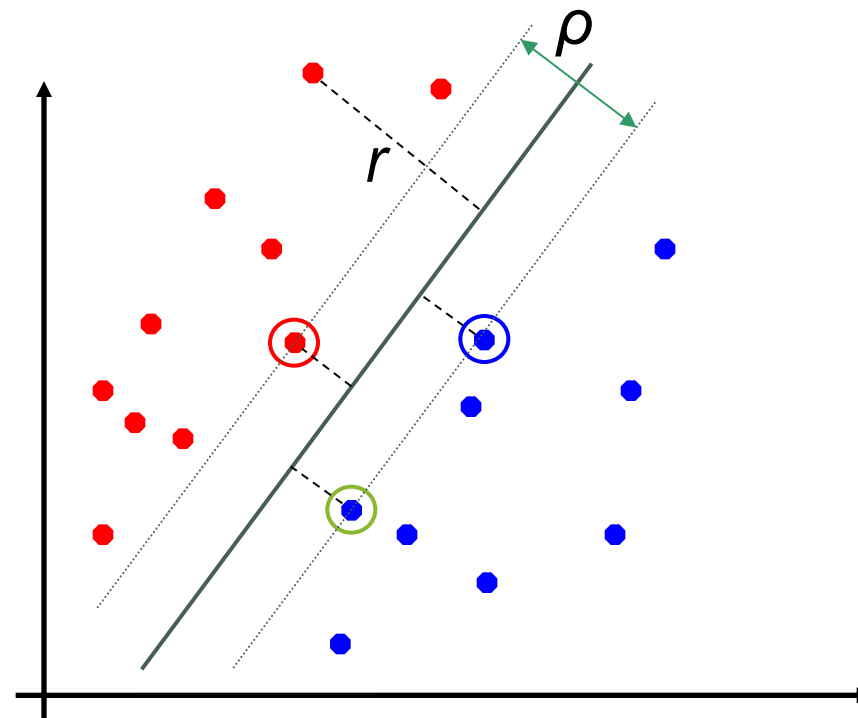
# Aprendizagem de Máquina

## Margem de Classificação

- Distância do exemplo  $x_i$  ao separador é

$$r = \frac{\mathbf{w}^T \mathbf{x}_i + b}{\|\mathbf{w}\|}$$

- Exemplos mais próximos ao hiperplano são **vetores de suporte**.
- **Margem**  $\rho$  do separador é a distância entre vetores de suporte de classes diferentes.



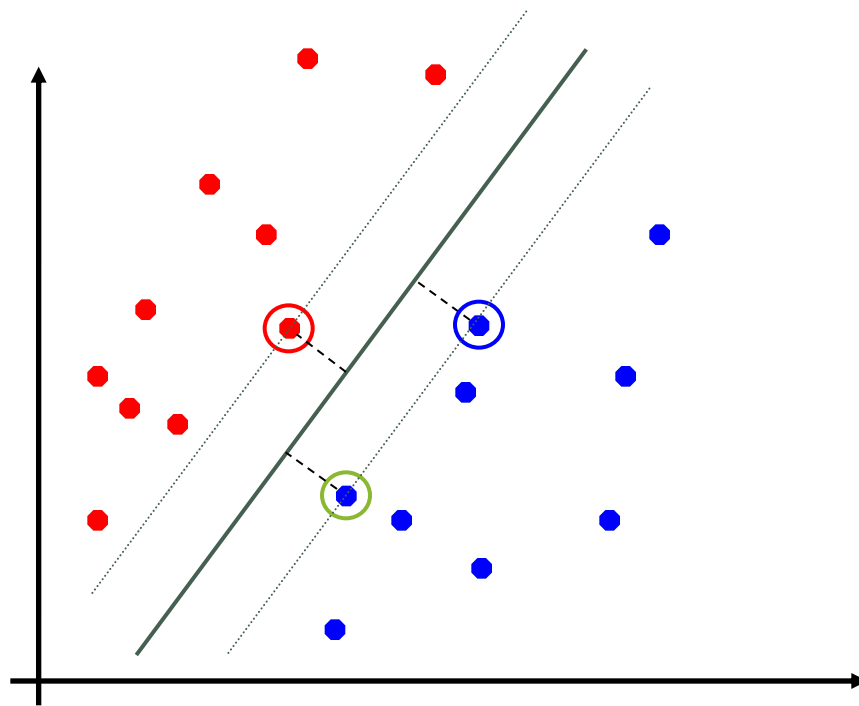
# Aprendizagem de Máquina

## Classificação com Máxima Margem

- Maximizar a margem é bom de acordo com a intuição e com a **teoria PAC**.
- Implica que só os vetores de suporte são importantes; outros exemplos de treinamento podem ser ignorados.

## Vetores de Suporte

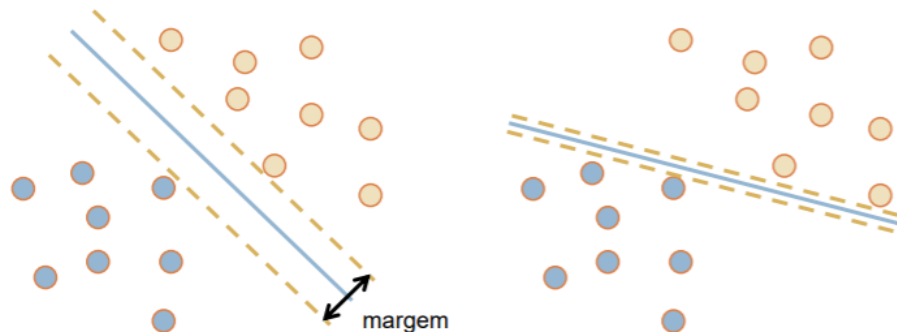
- Servem para definir qual será o hiperplano.
- São encontrados durante a fase de treinamento.



# Aprendizagem de Máquina

## Máquinas de Vetores Suporte

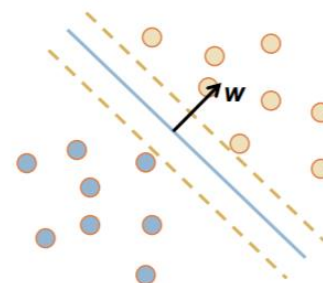
- O algoritmo SVM constrói os denominadores classificadores lineares, que separam o conjunto de dados por meio de um **hiperplano**.
- Conjunto de dados contendo somente duas classes (-1 e +1)
- Encontrar o hiperplano que maximiza a margem do limiar de decisão



# Aprendizagem de Máquina

## Máquinas de Vetores Suporte

- O hiperplano de separação é dado pela equação:  $f(x) = wx + b = 0$   
Onde  $w$  é o vetor de pesos (mesma dimensão das amostras) perpendicular ao hiperplano de separação e  $b$  é um escalar.
- A equação divide o espaço duas regiões
  - $wx + b > 0$
  - $wx + b < 0$
- Apenas o sinal é necessário para fazer a classificação



$$y(x) = \begin{cases} +1, & \text{se } wx + b > 0 \\ -1, & \text{se } wx + b < 0 \end{cases}$$

- Selecionar  $w$  e  $b$  de forma que os exemplos mais próximos ao hiperplano satisfaçam
  - $|wx + b| = 1$
- Assim temos que:

$$\begin{cases} wx + b \geq +1 & \text{se } y = +1 \\ wx + b \leq -1 & \text{se } y = -1 \end{cases}$$

# Aprendizagem de Máquina

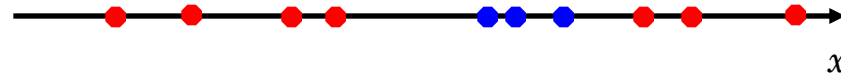
## Máquinas de Vetores Suporte

- A aplicação de um método puramente linear para classificar um conjunto de dados pode sofrer com **dois problemas** bastante comuns:
  - *Outliers*
  - Exemplos rotulados erroneamente
- Em alguns problemas não é possível separar as classes linearmente mesmo utilizando a margem de folga.
- Na realidade, a grande maioria dos problemas reais não são separáveis linearmente.
- O que fazer?

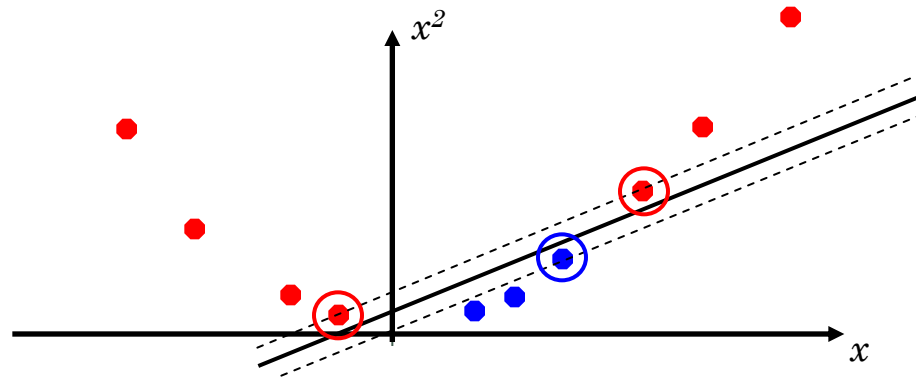
# Aprendizagem de Máquina

## Máquinas de Vetores Suporte Não-linear

- O que fazer quando os dados não são linearmente separáveis?



- A abordagem utilizada pelo SVM para resolver esse tipo de problema consiste em mapear os dados para um espaço de dimensão maior:

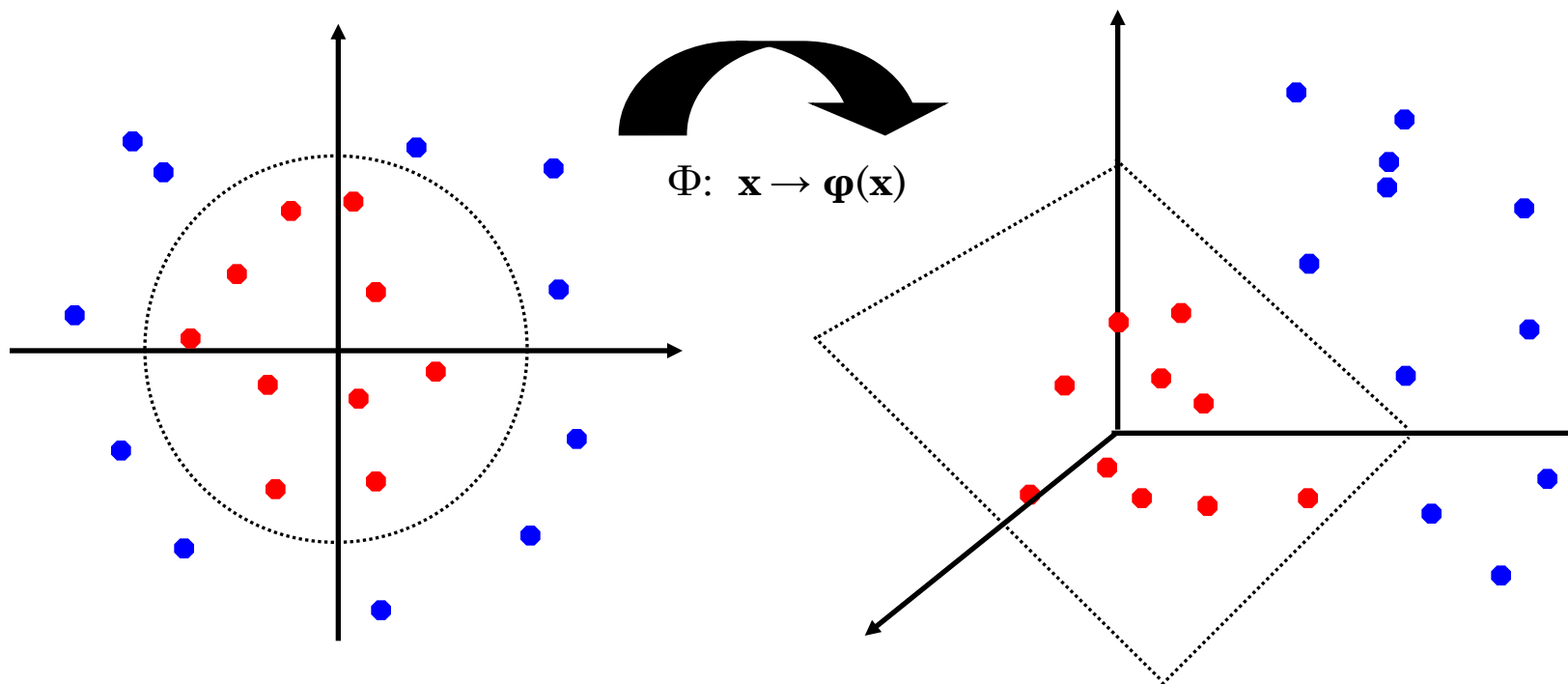




# Aprendizagem de Máquina

## Máquinas de Vetores Suporte Não-linear

- O espaço de atributos original pode ser mapeado em um espaço de atributos de dimensão maior onde o conjunto de treinamento é linearmente separável:

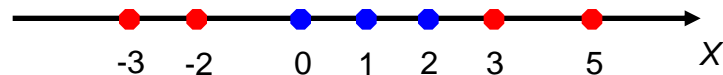


# Aprendizagem de Máquina

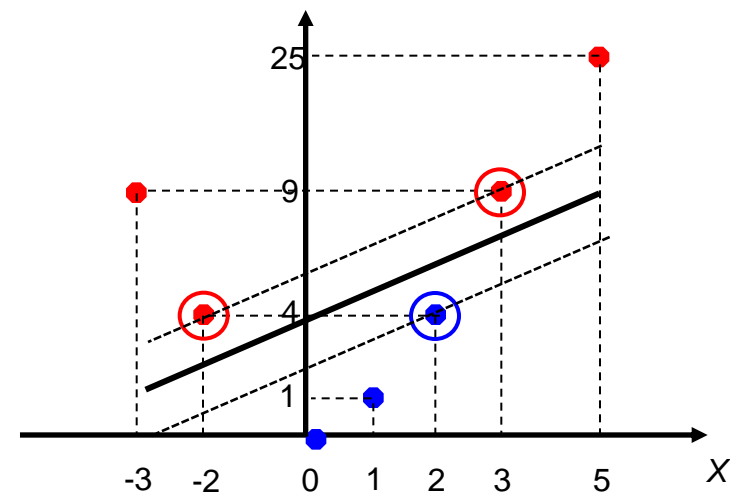
## Máquinas de Vetores Suporte Não-linear

Exemplo:

- Considerando o seguinte conjunto de exemplos de treinamento que não são linearmente separáveis:



- Elevando para uma dimensão linearmente separável ( $\mathbb{R}^1 \rightarrow \mathbb{R}^2$ ):



- Kernel:**  $\varphi(x) = (x, x^2)$

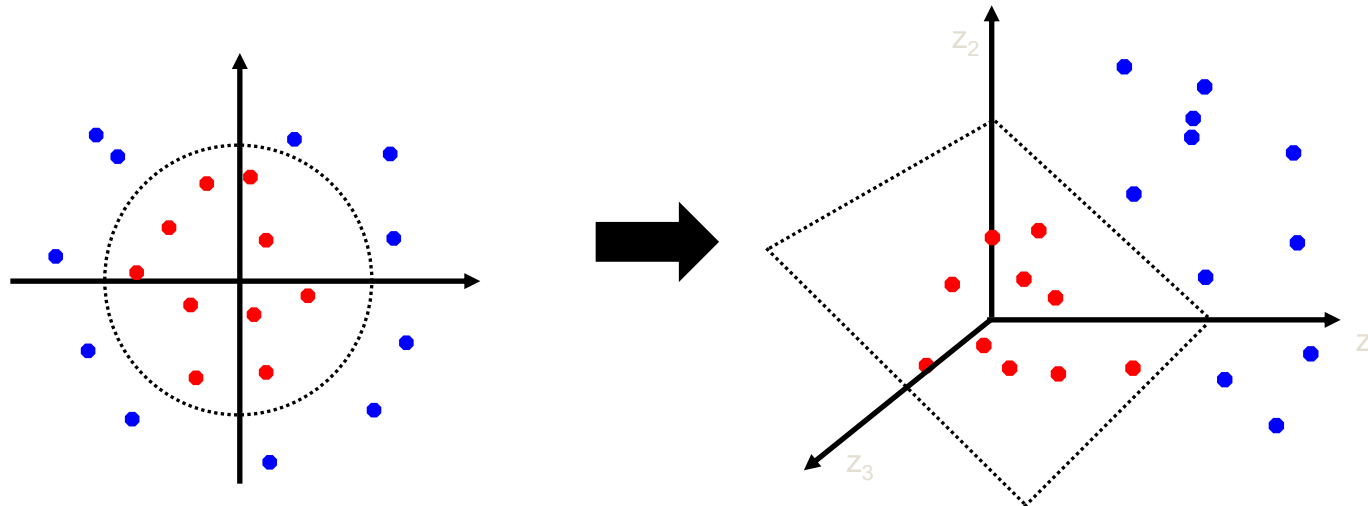
# Aprendizagem de Máquina

## Máquinas de Vetores Suporte Não-linear

Exemplo:

- A mesma metodologia pode ser aplicada em um espaço 2D de características ( $\mathbb{R}^2 \rightarrow \mathbb{R}^3$ ).
- A única diferença é a necessidade de uma nova função de kernel. Um exemplo de função de kernel aplicável nesse caso seria:

$$\phi(x_1, x_2) \rightarrow (z_1, z_2, z_3) = (x_1^2, \sqrt{2x_1x_2}, x_2^2)$$

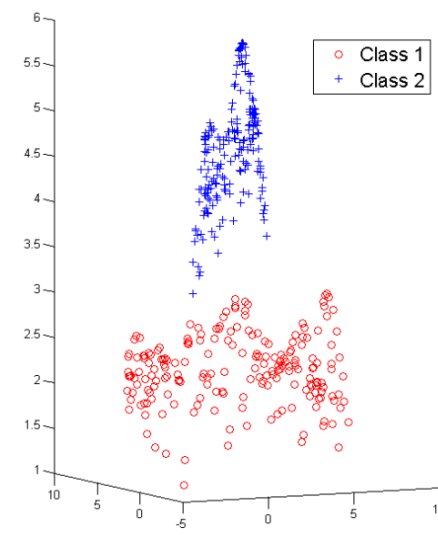
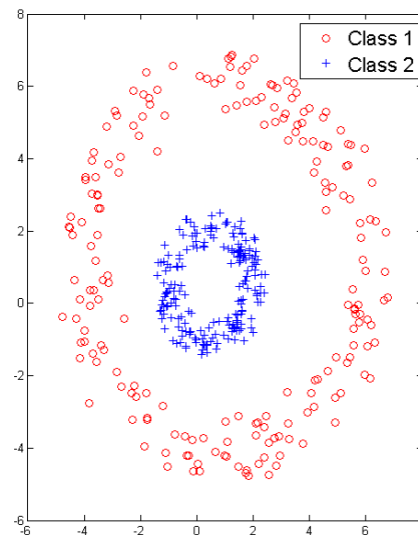


# Aprendizagem de Máquina

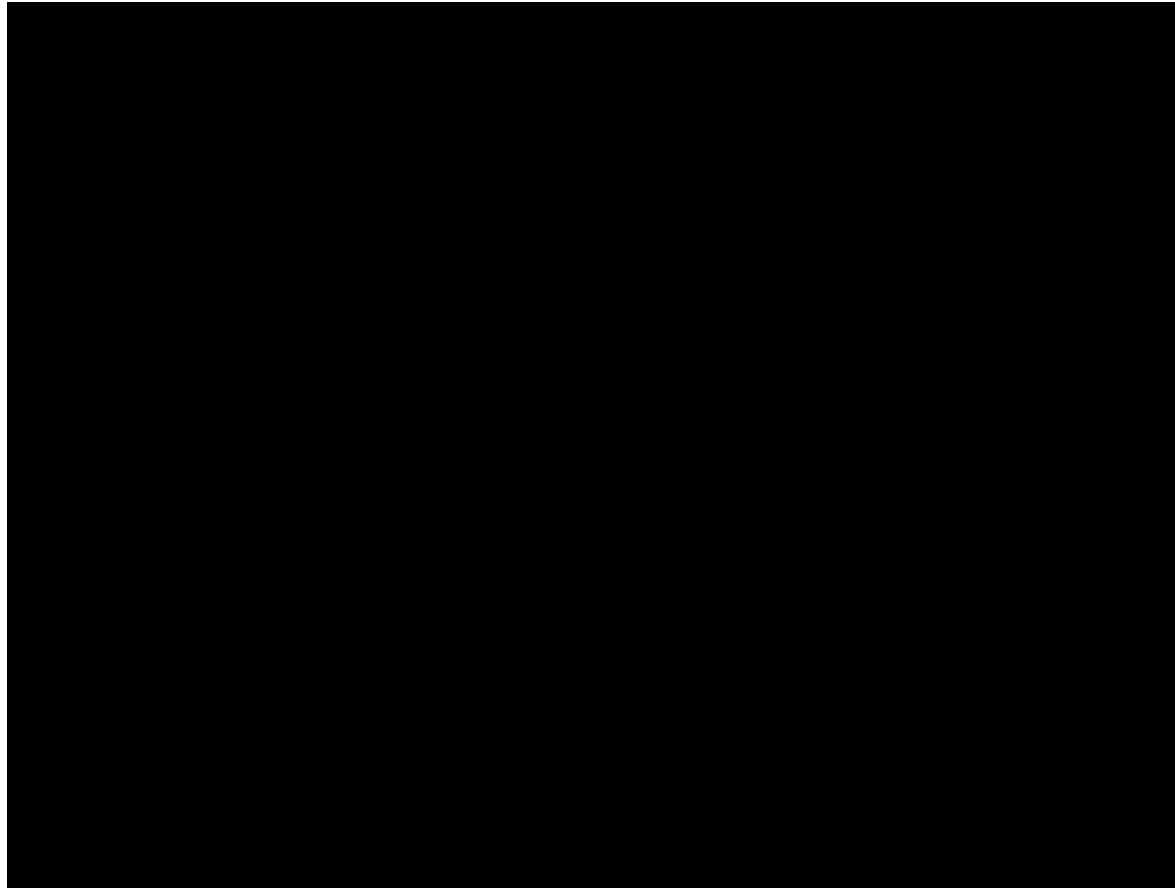
## Máquinas de Vetores Suporte Não-linear

- Funções Kernel

Kernel	Função $\phi(x_i, x_j)$
Polinomial	$(\delta(x_i \cdot x_j) + k)^d$
Gaussiano	$\exp(-\sigma \ x_i - x_j\ ^2)$
Sigmoidal	$\tanh(\delta(x_i \cdot x_j) + k)$



# Polynomial Kernel



# Aprendizagem de Máquina

## Máquinas de Vetores Suporte Não-linear

- O SVM foi originalmente concebido para lidar com **classificações binárias**.
- Entretanto, a maior parte dos problemas reais requerem **múltiplas classes**.
- Para se utilizar uma SVM para classificar múltiplas classes é necessário transformar o problema multi-classe em vários problemas de classes binárias

# Aprendizagem de Máquina

## Vantagens:

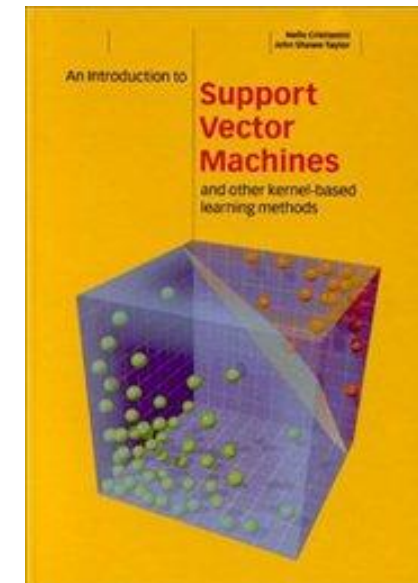
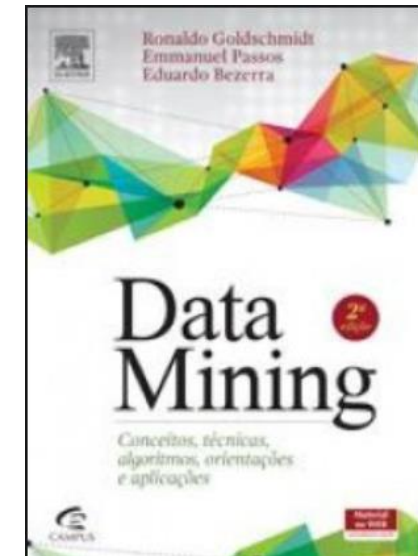
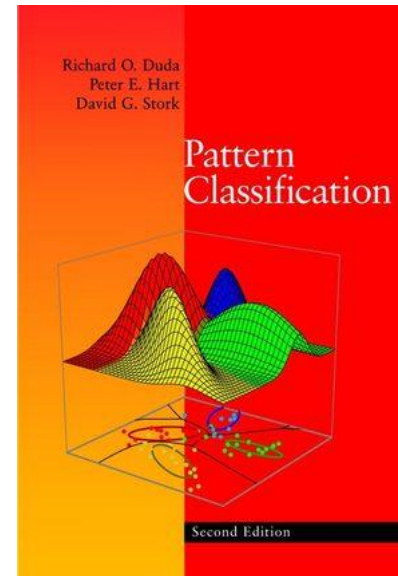
- Consegue lidar bem com grandes conjuntos de exemplos.
- Trata bem dados de alta dimensão.
- O processo de classificação é rápido.

## Desvantagens:

- É necessário definir um bom Kernel.
- O tempo de treinamento pode ser bem longo dependendo do número de exemplos e dimensionalidade dos dados.

# Bibliografia

- Mitchell, T. **Machine Learning** (1997). McGraw–Hill Science/Engineering/Math.
- Duda, R., Hart, P., Stork, D., **Pattern Classification** (2000). John Wiley & Sons.
- Cristianini, N., Shawe-Taylor, J., **An Introduction to Support Vector Machines and Other Kernel-based Learning Methods** (2000) Cambridge University Press.
- Goldschmidt, R. Passos, E. Bezerra, E. **Data Mining** (2015). Elsevier.





# Extras...

