

Introdução ao Aprendizado de Máquina

Lucas Gonçalves de Moura Leite

Aula de hoje

- ▶ **Aprendizado Supervisionado**
 - ▶ Naive Bayes
 - ▶ Random Forest
 - ▶ Gradient Boosted Decision Tree
 - ▶ Redes Neurais
- ▶ **Aprendizado Não-supervisionado**
 - ▶ Redução de dimensionalidade
 - ▶ Clustering



Naive Bayes

Naive Bayes

- ▶ Assume que os atributos são descorrelacionados
- ▶ Hipótese simplifica o modelo
 - ▶ Custo de treinamento
- ▶ Em geral possui desempenho pior que outros classificadores
- ▶ Pode ter bom desempenho para algumas aplicações
- ▶ Calcular a probabilidade de um determinado dado pertencer a uma das classes

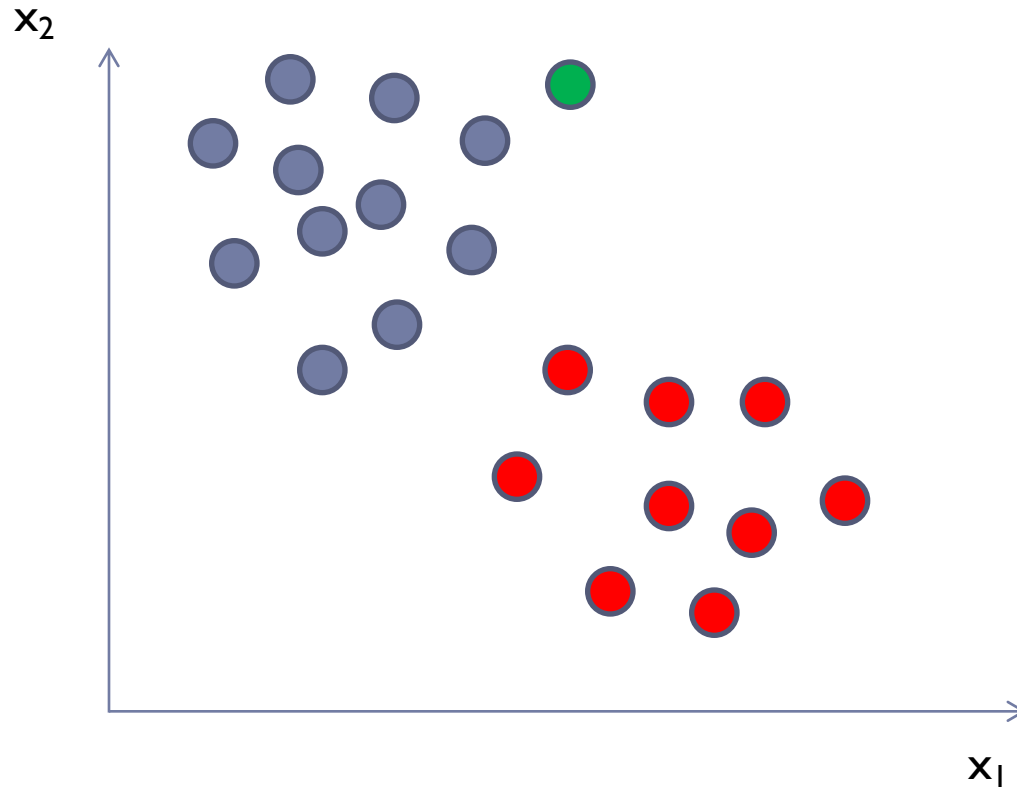


Naive Bayes – Scikit-learn

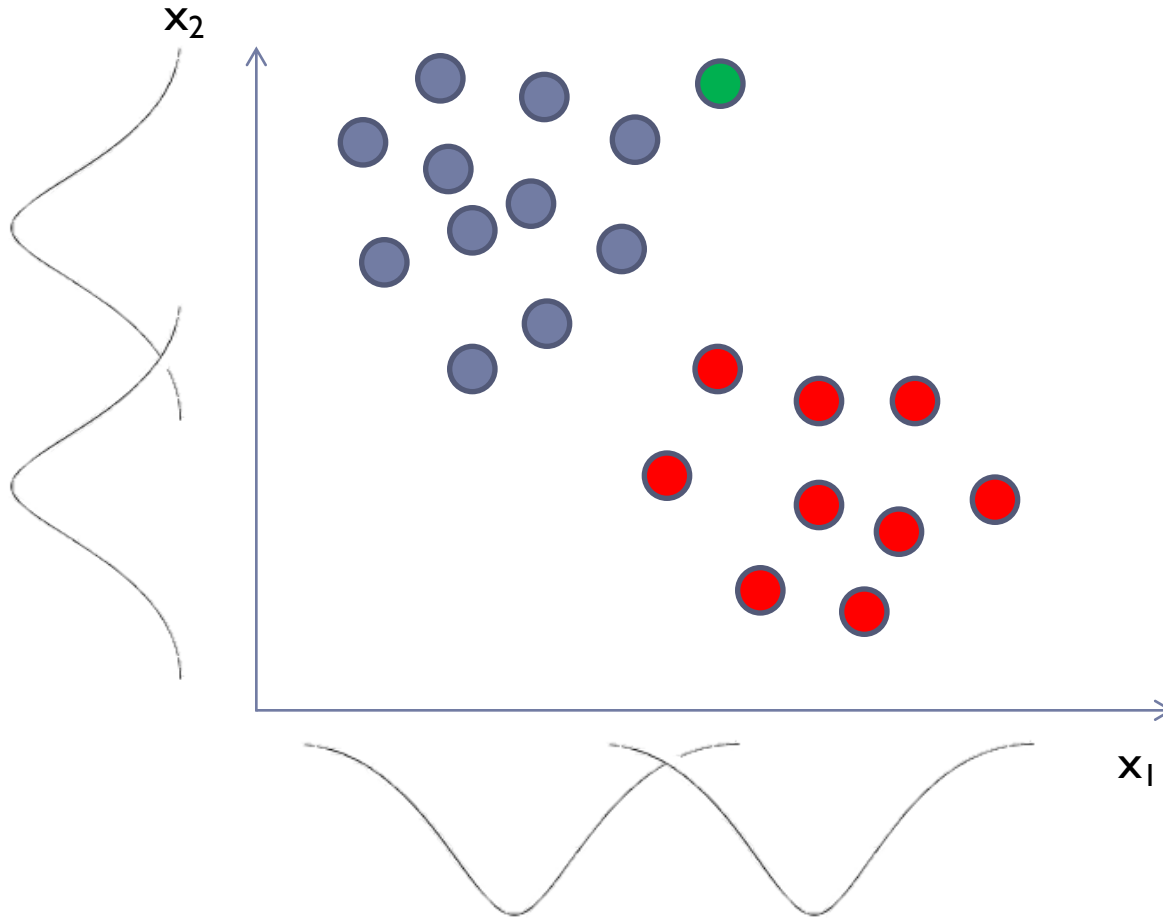
- ▶ **Bernoulli**
 - ▶ Atributos binários
- ▶ **Multinomial**
 - ▶ Atributos discretos
- ▶ **Gaussian**
 - ▶ Atributos reais



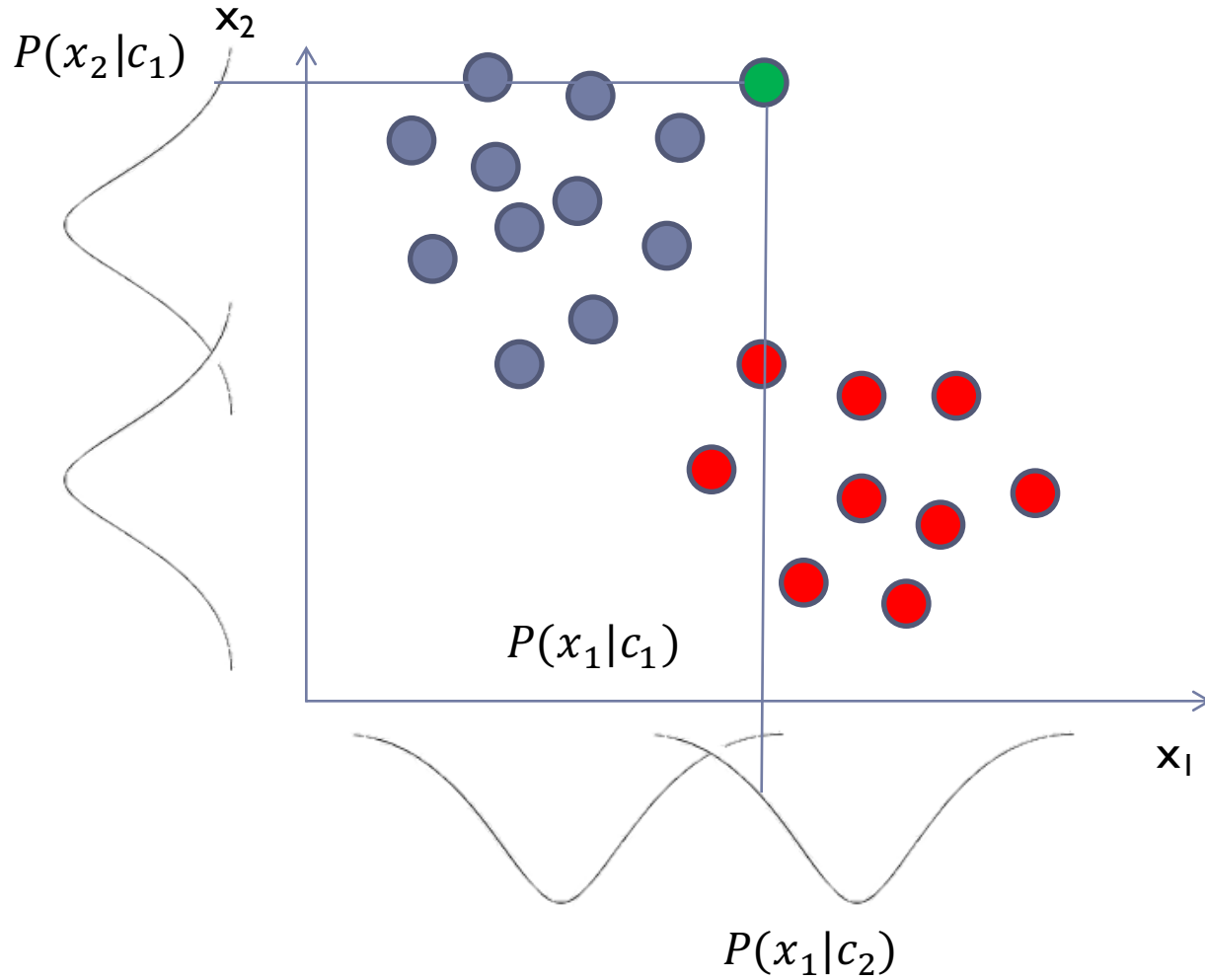
Gaussian Naive Bayes



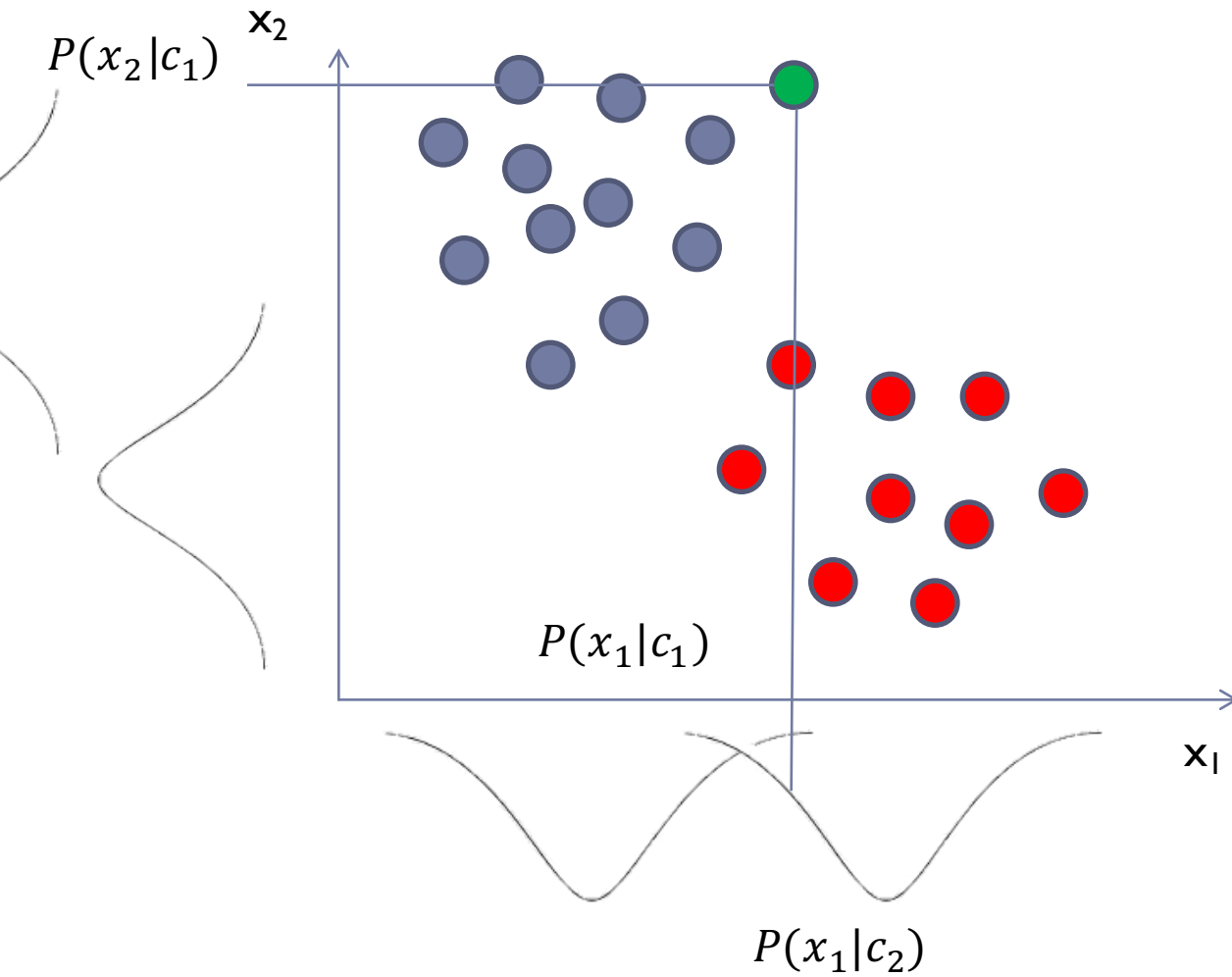
Gaussian Naive Bayes



Gaussian Naive Bayes



Gaussian Naive Bayes



$$P(c_1|x) = \prod_{i=1}^N p(x_i|c_1)$$
$$P(c_2|x) = \prod_{i=1}^N p(x_i|c_2)$$

Naive Bayes

- ▶ Criar modelo independente para cada atributo
 - ▶ Custo computacional baixo
- ▶ Classificador Linear/Quadrático
- ▶ Sem hiperparâmetros
- ▶ Muito utilizado para dados com alta dimensão
 - ▶ Classificação de textos



Exercício

- ▶ Usar o Gaussian Naive Bayes para os dados breast cancer

```
from sklearn.naive_bayes import GaussianNB
X_train, X_test, y_train, y_test = train_test_split(X_C2, y_C2,
                                                    random_state=0)

nbclf = GaussianNB().fit(X_train, y_train)
```



Random Forest

Ensemble Models (Comitê de maquinas)

- ▶ Foi demonstrado que a combinação de modelos tende a gerar melhores resultados que os seus componentes individuais.
- ▶ Diferentes modelos cometem erros diferentes (overfitting de formas diferentes)
- ▶ A combinação alivia o efeito do overfitting
 - ▶ Os modelos devem ser diversos



Random Forest

- ▶ Ensemble de árvores
- ▶ Muito usada
 - ▶ Excelentes resultados em muitas aplicações
- ▶ Modulo `sklearn.ensemble`
 - ▶ `RandomForestClassifier`
 - ▶ `RandomForestRegressor`
- ▶ Como tornar as árvores diversas ?



Random Forest

Dados originais

fruit_label	fruit_name
1	Apple
1	Apple
1	Apple
1	Apple
2	Mandarin
...	...
3	Orange
...	...
4	Lemon
4	Lemon
4	Lemon
4	Lemon
4	Lemon

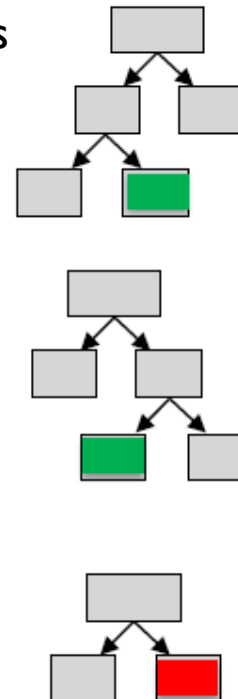
N conjuntos
aleatórios
(bootstrap)

1	Apple
1	Apple
2	Mandarin
...	...
3	Orange
...	...
4	Lemon
4	Lemon
4	Lemon
4	Lemon
4	Lemon

fruit_label	fruit_name
1	Apple
1	Apple
1	Apple
1	Apple
2	Mandarin
...	...
3	Orange
...	...
4	Lemon
4	Lemon
4	Lemon
4	Lemon
4	Lemon

fruit_label	fruit_name
1	Apple
1	Apple
1	Apple
1	Apple
2	Mandarin
...	...
3	Orange
...	...
4	Lemon
4	Lemon
4	Lemon
4	Lemon
4	Lemon

Subconjunto
aleatório dos
atributos



Ensemble



n_estimator

max_features

Random Forest

- ▶ **Dados**
 - ▶ Bootstrap
- ▶ **Atributos**
 - ▶ Seleção aleatória
- ▶ **Resultado**
 - ▶ Regressão
 - ▶ Média da saídas das árvores
 - ▶ Classificação
 - ▶ Probabilidade fornecida por cada árvore
 - ▶ Probabilidade média para cada classe



Random Forest

▶ Vantagens

- ▶ Boa performance
- ▶ Não é muito sensível a escolha dos parâmetros
- ▶ Facilmente paralelizável

▶ Desvantagens

- ▶ Sem interpretabilidade
- ▶ Desempenho deficiente para dados com muitos atributos



Exercício

- ▶ Usar o Random Forest para os dados breast cancer

```
from sklearn.ensemble import RandomForestClassifier  
clf = RandomForestClassifier(max_features = 8, random_state = 0)
```



Gradient Boosted Decision Tree

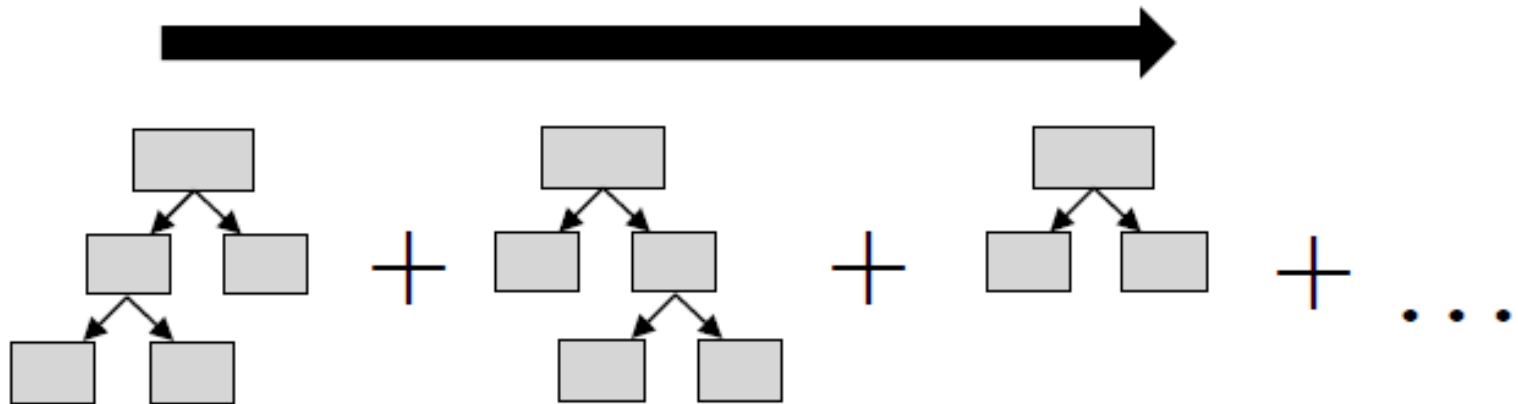
Gradient Boosted Decision Tree

- ▶ **Ensemble model**
- ▶ **Random Forest**
 - ▶ Árvores em paralelo
- ▶ **GBDT**
 - ▶ Árvores em série



Gradient Boosted Decision Tree

- ▶ Weak Learners
- ▶ Cada nova árvore busca corrigir os erros da anterior



Gradient Boosted Decision Tree

▶ Parâmetros

- ▶ `learning_rate`: define o quanto uma nova árvore tentará corrigir os erros da anterior
- ▶ `max_depth`: profundidade da árvore
- ▶ `n_estimators`: numero de árvores



Gradient Boosted Decision Tree

▶ Vantagens

- ▶ Boa performance
- ▶ Não é muito sensível a escolha dos parâmetros
- ▶ Uso do modelo tem baixo custo computacional

▶ Desvantagens

- ▶ Sem interpretabilidade
- ▶ Treinamento é custoso
- ▶ Desempenho deficiente para dados com muitos atributos



Exercício

- ▶ Usar o GBDT para os dados breast cancer
- ▶ Usar os parâmetros
 - ▶ learning_rate=0.1,max_depth=3
 - ▶ learning_rate=0.1,max_depth=2

```
from sklearn.ensemble import GradientBoostingClassifier  
clf = GradientBoostingClassifier(random_state = 0)
```



Redes Neurais

Redes Neurais

- ▶ Funcionamento inspirado no neurônio biológico.
- ▶ Tarefas de Aprendizado de Máquina
 - ▶ Classificação
 - ▶ Regressão



Neurônio Biológico x Neurônio Artificial

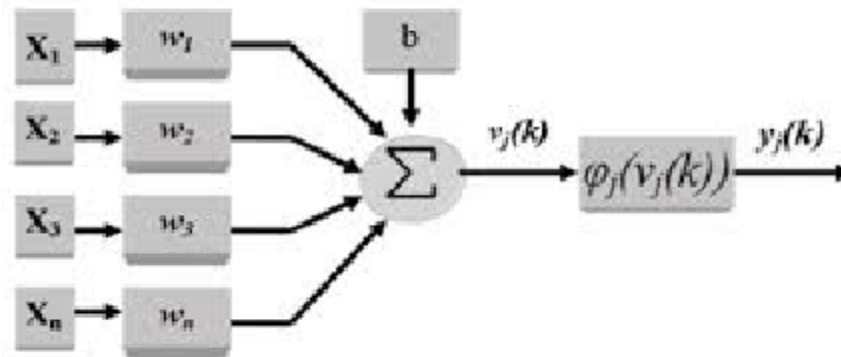


Figura 2: Representação do neurônio artificial.

Neurônio Biológico x Neurônio Artificial



McCulloch-Pitts

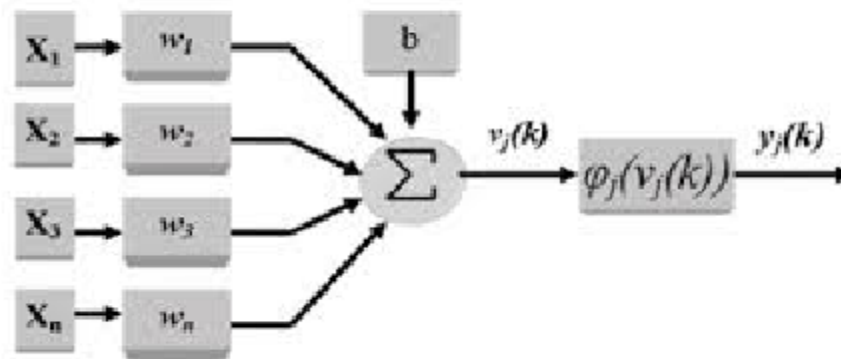


Figura 2: Representação do neurônio artificial.

Modelo de McCulloch-Pitts

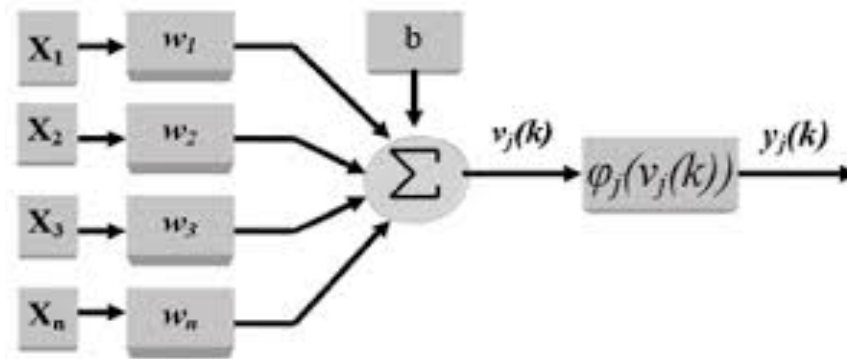


Figura 2: Representação do neurônio artificial.

$$v_j = w_1 x_1 + w_2 x_2 + \dots - b$$

$$\phi(v_j) = 1 \text{ se } v_j > 0$$

$$\phi(v_j) = 0 \text{ se } v_j < 0$$

Perceptron

Modelo + Regra de aprendizado

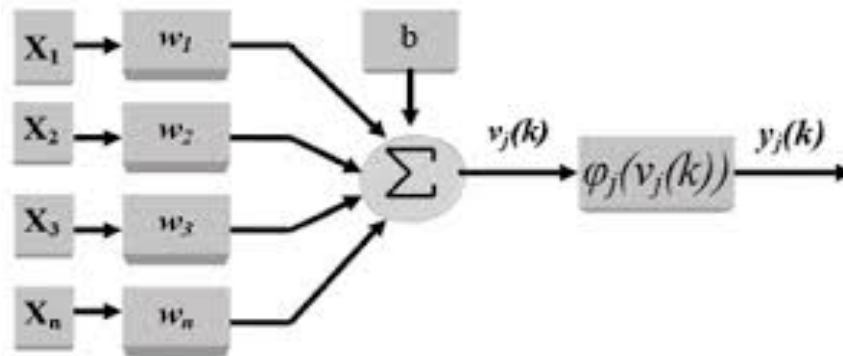
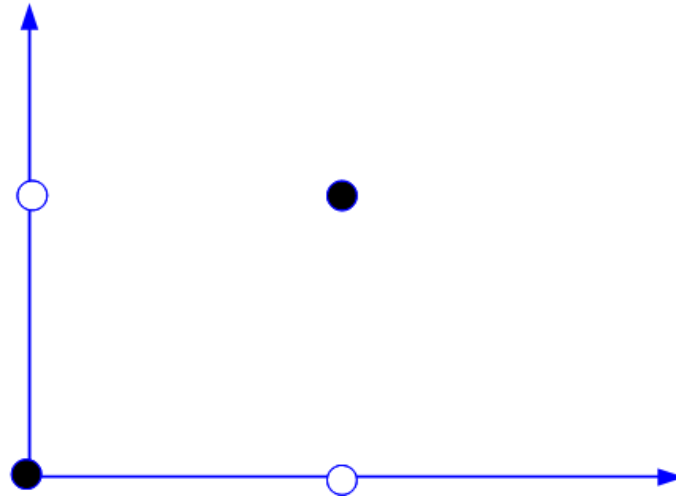


Figura 2: Representação do neurônio artificial.

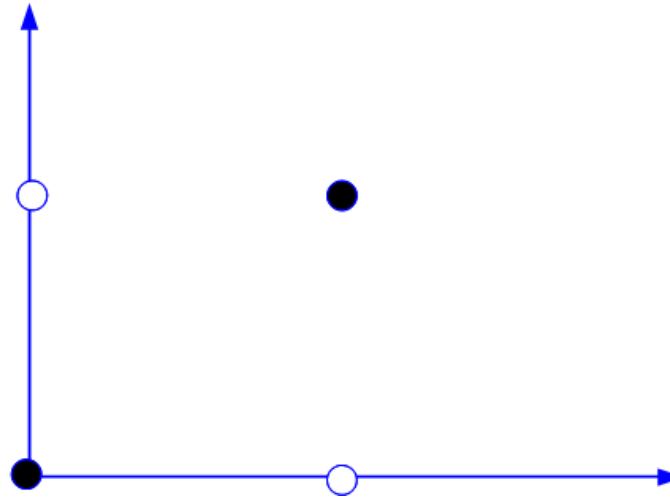
Perceptron

Entrada (x)	Saída (y)
0,0	0
0,1	1
1,0	1
1,1	0



Perceptron

Entrada (x)	Saída (y)
0,0	0
0,1	1
1,0	1
1,1	0



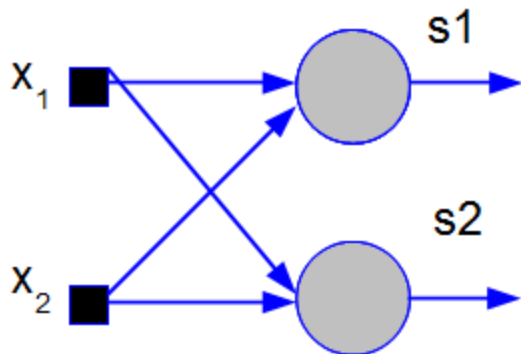
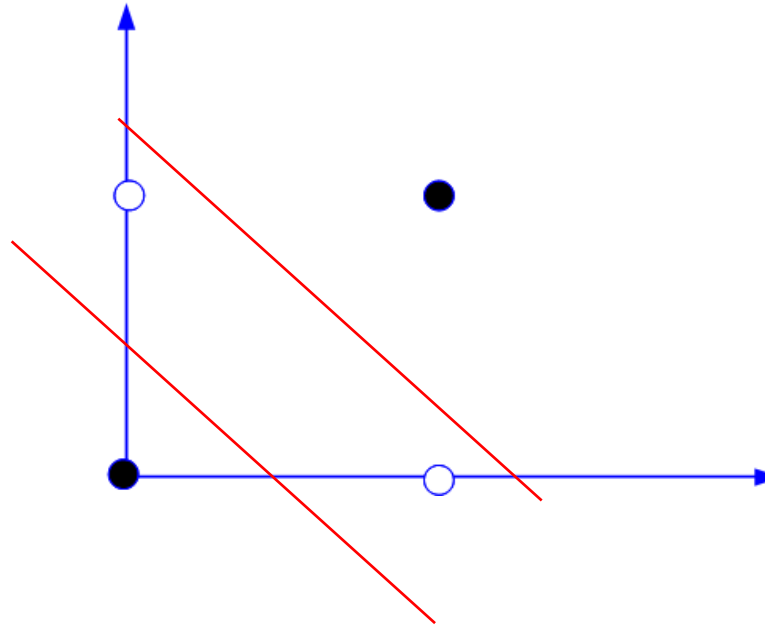
x_1 ■

x_2 ■



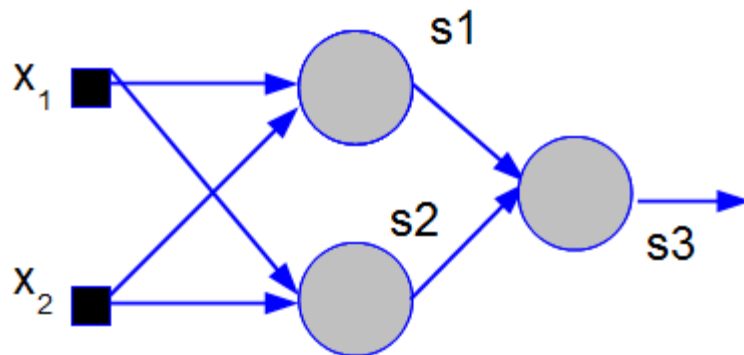
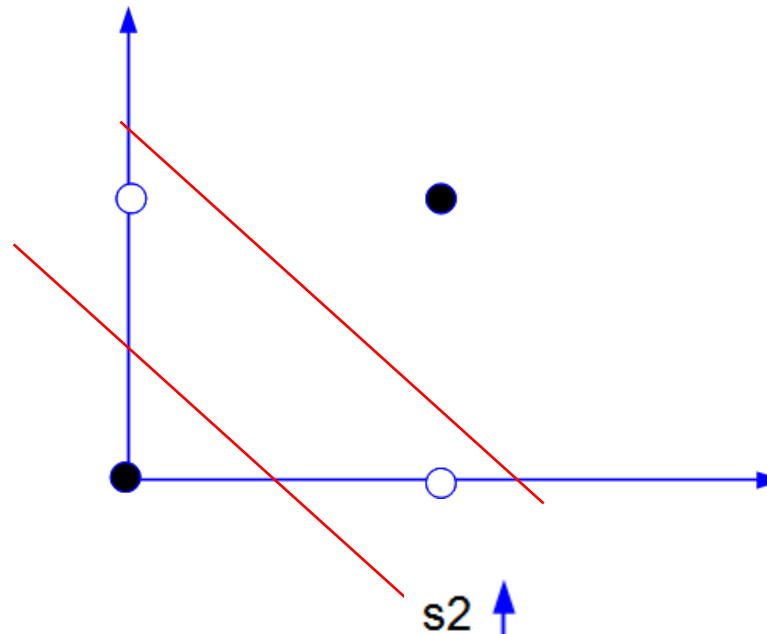
Perceptron

Entrada (x)	Saída (y)
0,0	0
0,1	1
1,0	1
1,1	0



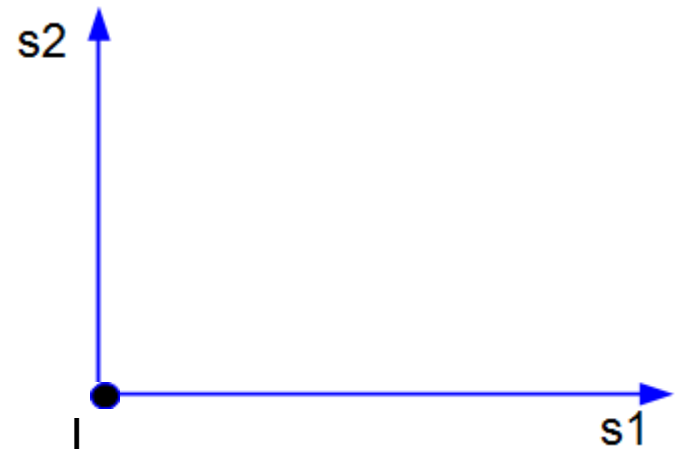
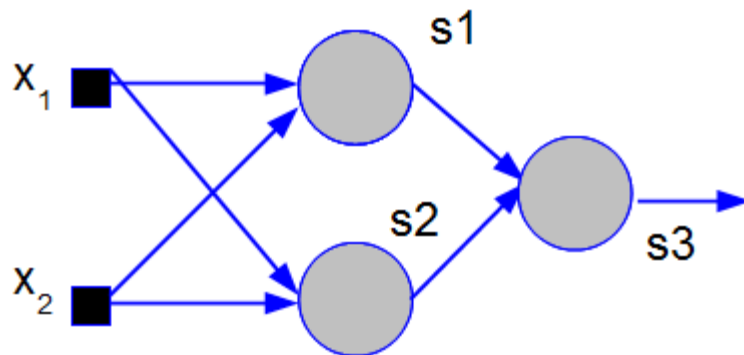
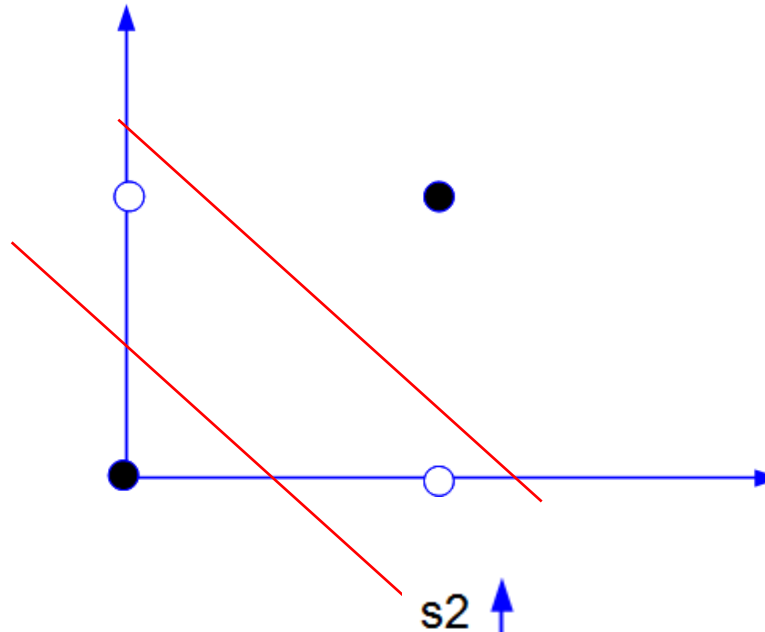
Perceptron

Entrada (x)	Saída (y)
0,0	0
0,1	1
1,0	1
1,1	0



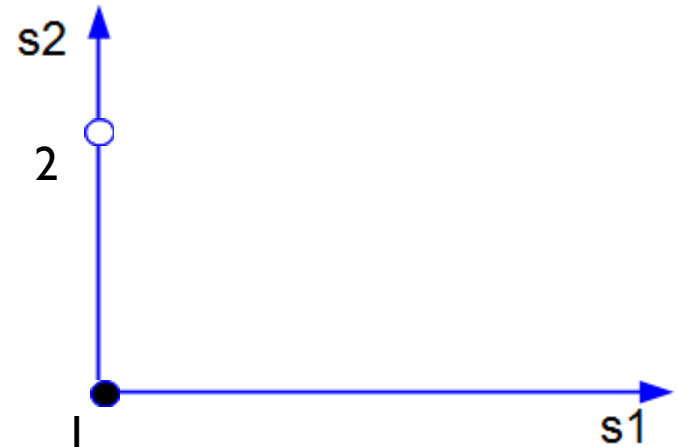
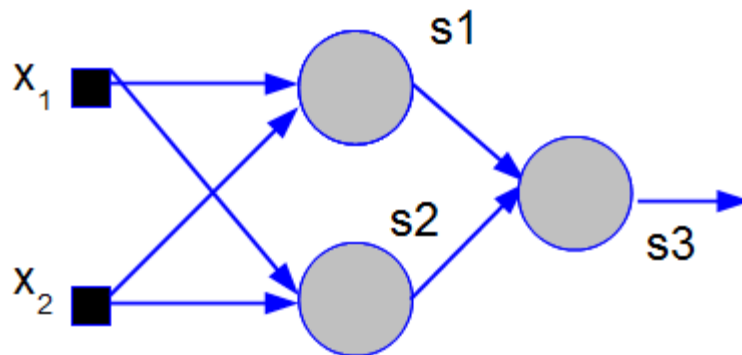
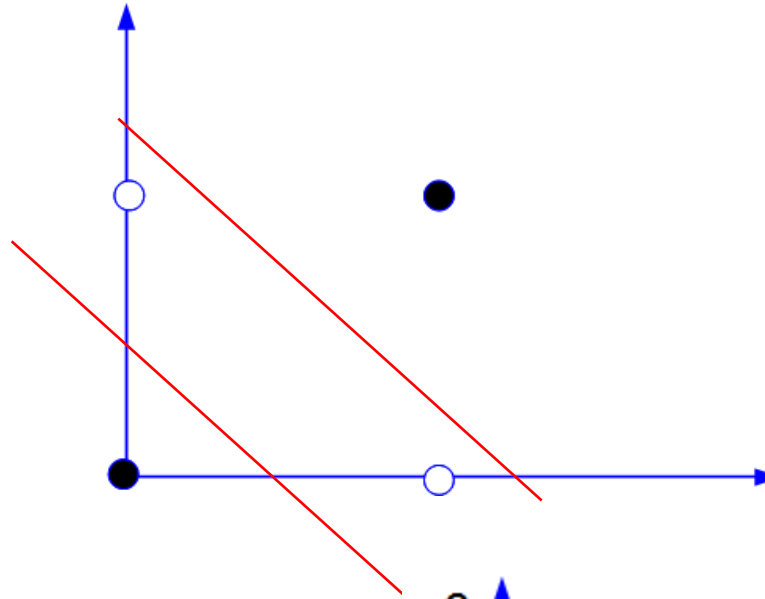
Perceptron

Entrada (x)	Saída (y)
0,0	0
0,1	1
1,0	1
1,1	0



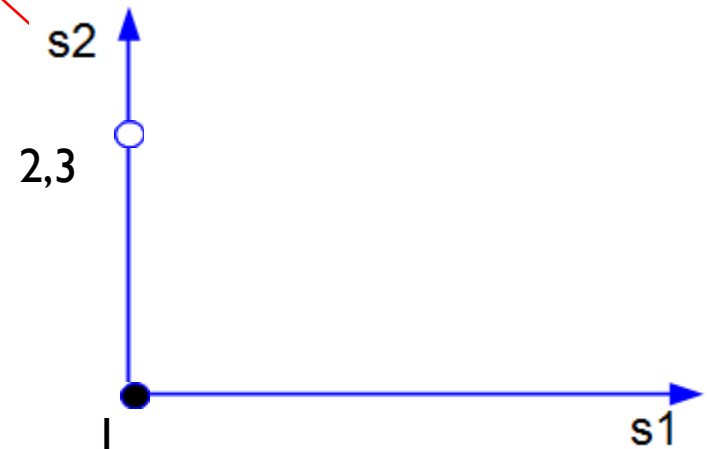
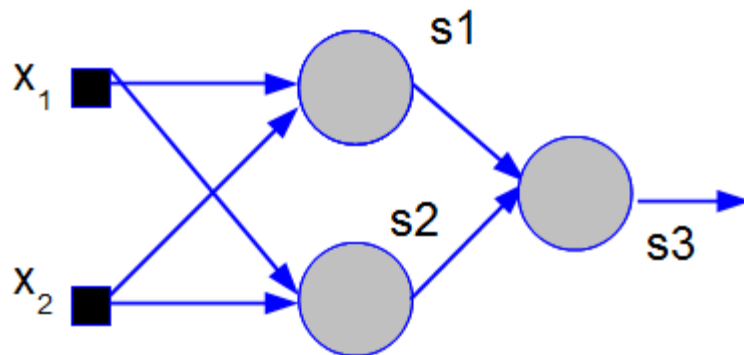
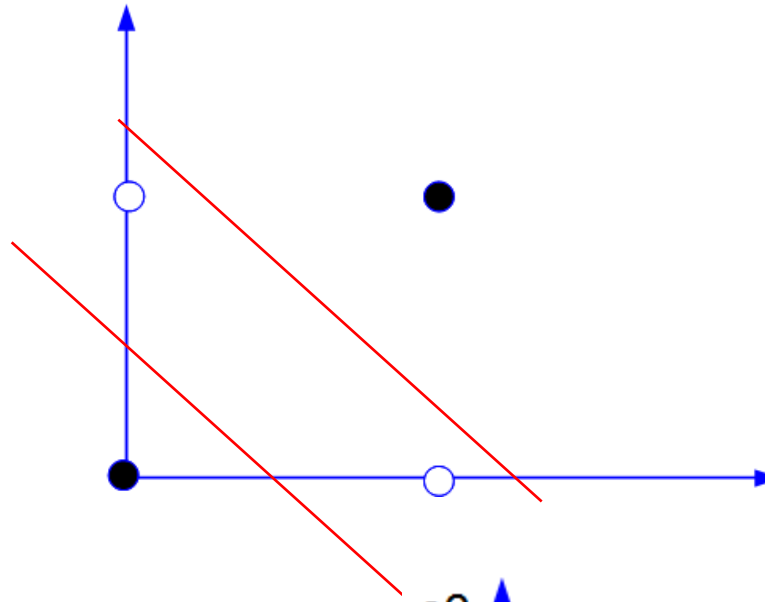
Perceptron

Entrada (x)	Saída (y)
0,0	0
0,1	1
1,0	1
1,1	0



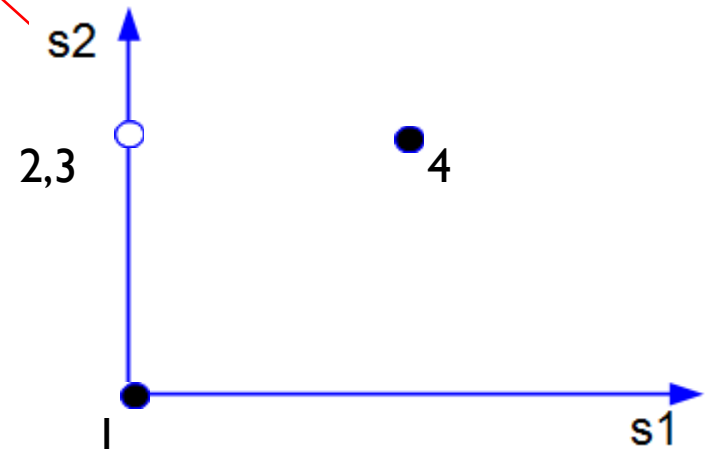
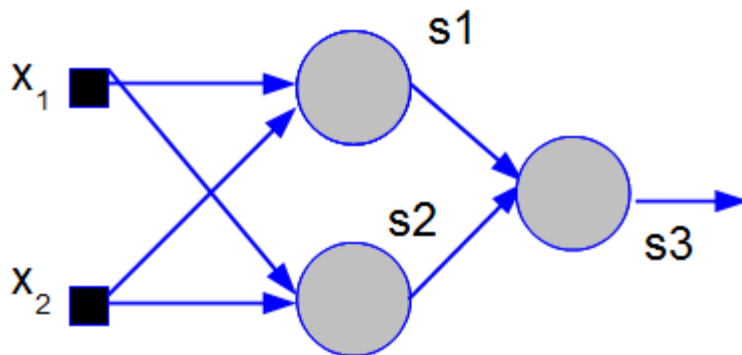
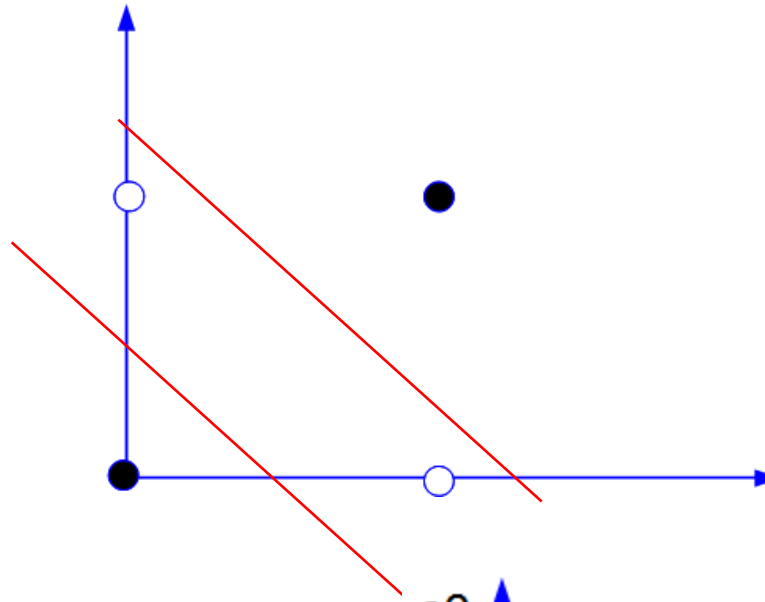
Perceptron

Entrada (x)	Saída (y)
0,0	0
0,1	1
1,0	1
1,1	0



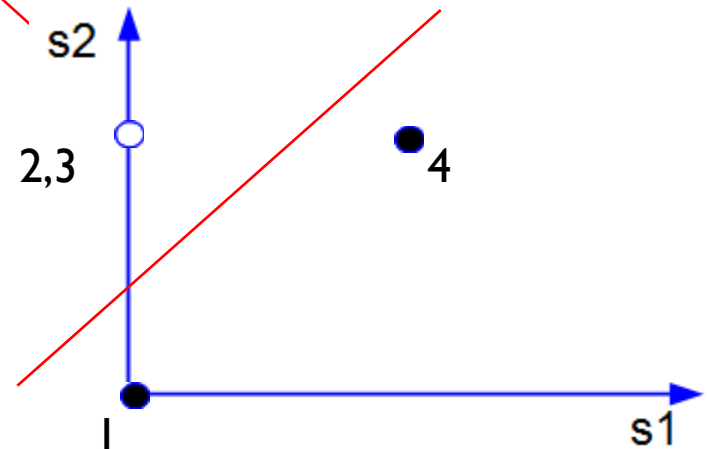
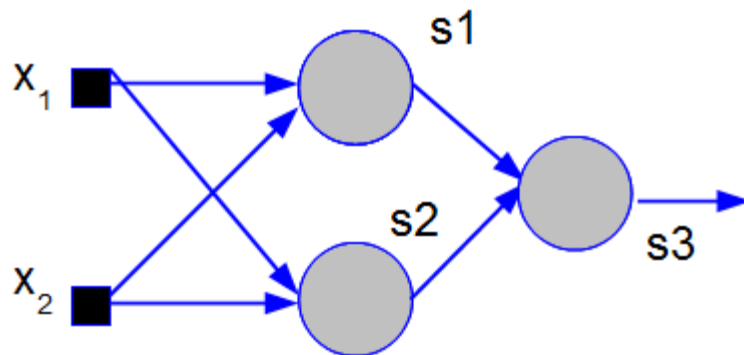
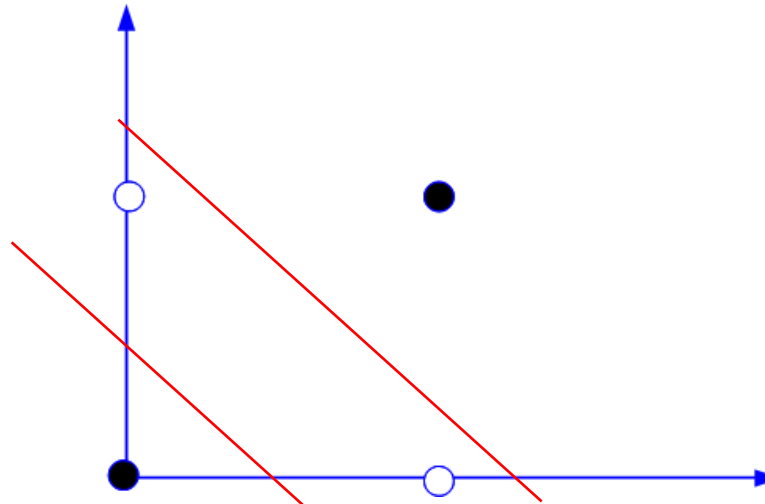
Perceptron

Entrada (x)	Saída (y)
0,0	0
0,1	1
1,0	1
1,1	0



Perceptron

Entrada (x)	Saída (y)
0,0	0
0,1	1
1,0	1
1,1	0



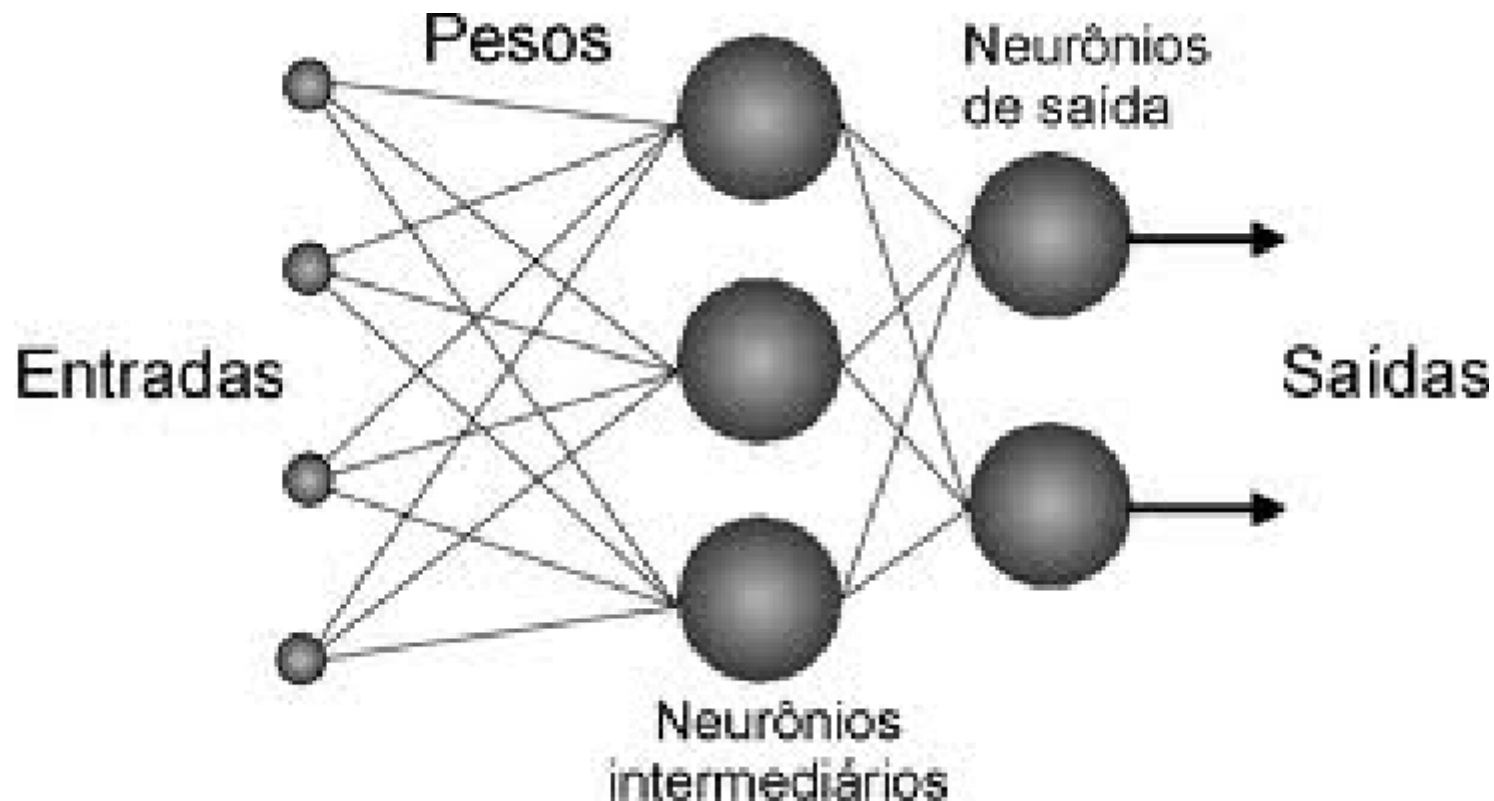
Perceptron de Múltiplas Camadas

- ▶ Rede MLP (MultiLayer Perceptron)
 - ▶ Problemas não linearmente separáveis



Redes MLP

- ▶ Redes com múltiplas camadas de neurônios artificiais



Redes MLP

► Scikit-learn

```
from sklearn.neural_network import MLPClassifier
nnclf = MLPClassifier(hidden_layer_sizes = 10, solver='lbfgs',
                      random_state = 0).fit(X_train, y_train)
```



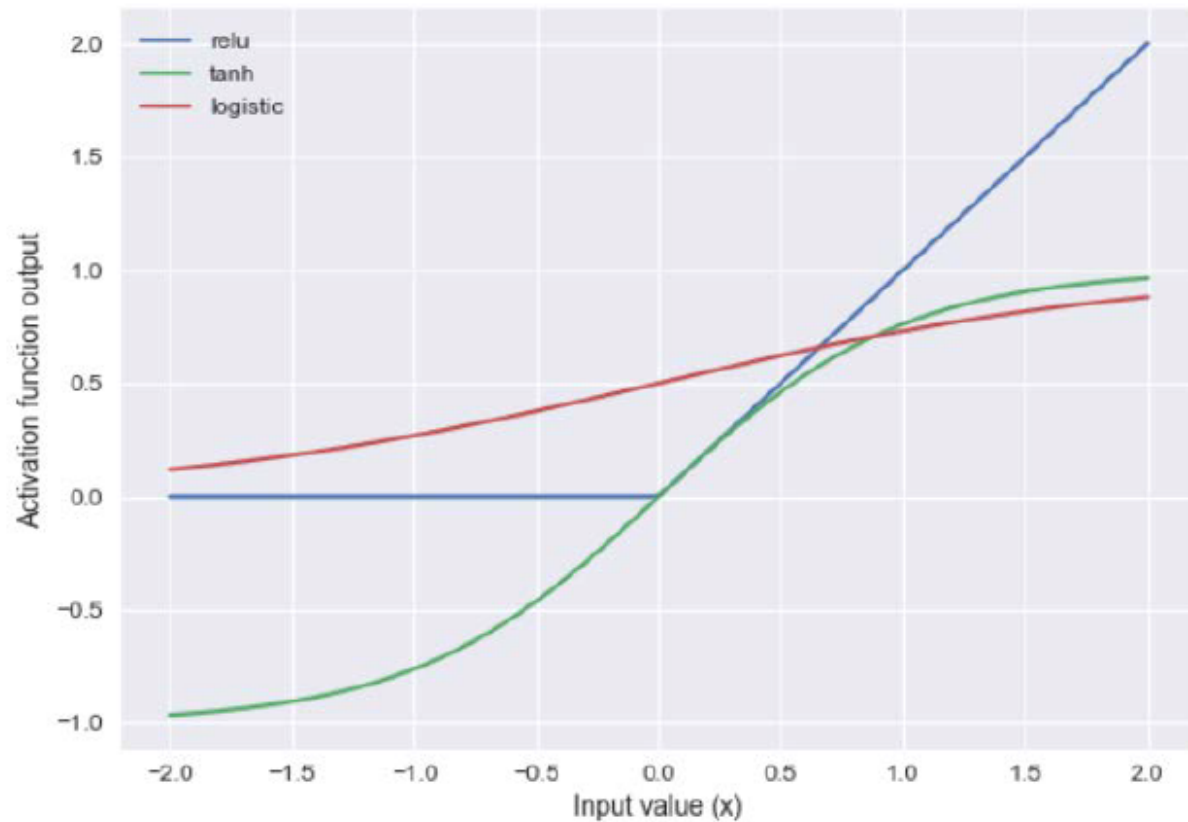
Parâmetros

- ▶ Função de ativação
- ▶ Número de camadas
- ▶ Número de neurônios por camada
- ▶ Parâmetro de regularização (alpha)

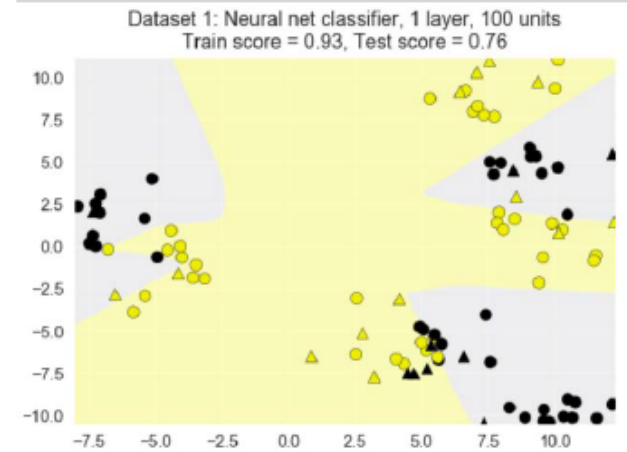
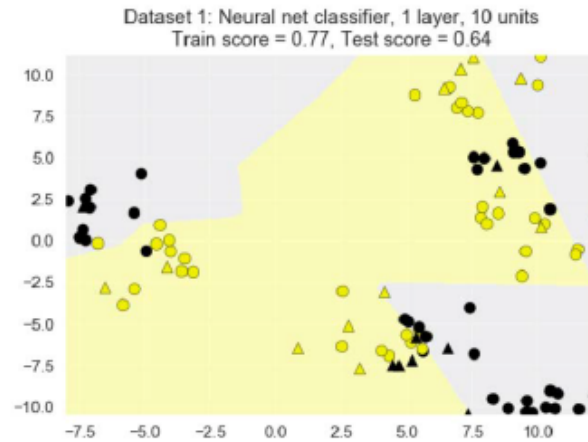
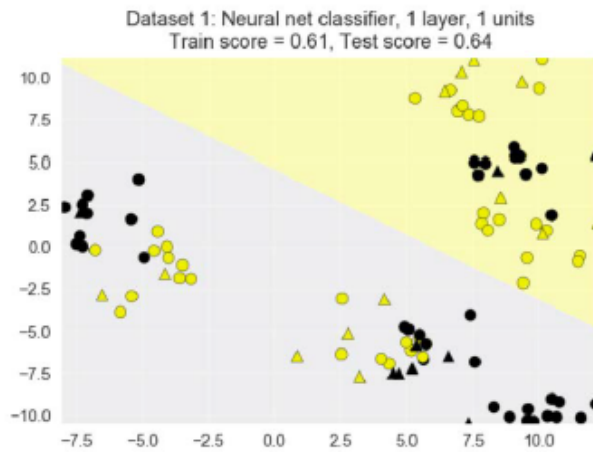


Funções de Ativação

► RELU - default

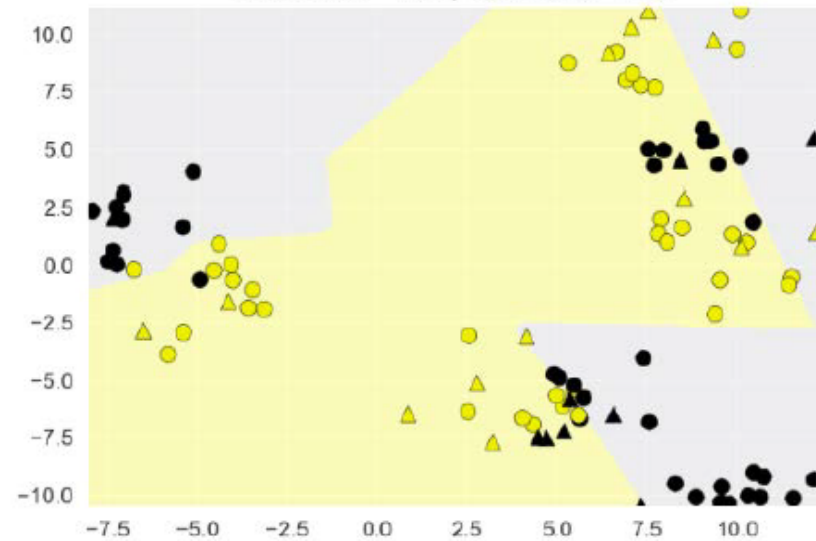


Efeito de numero de neurônios

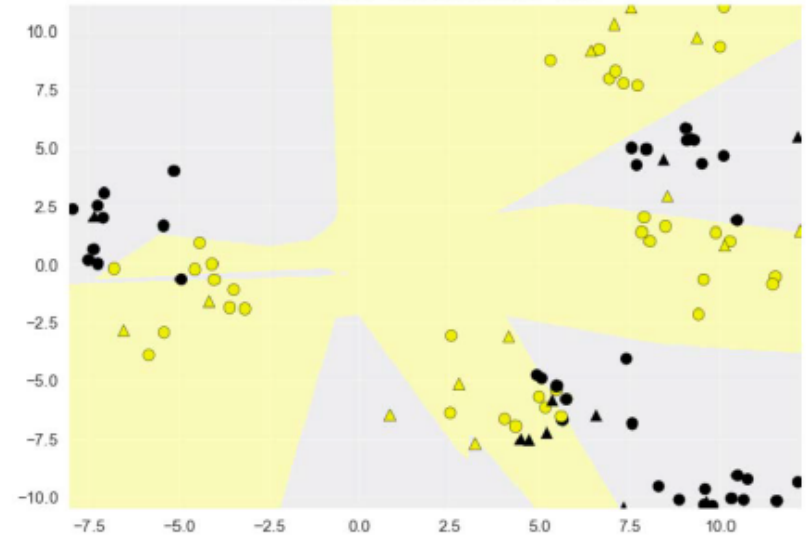


Efeito de numero de camadas

Dataset 1: Neural net classifier, 1 layer, 10 units
Train score = 0.77, Test score = 0.64

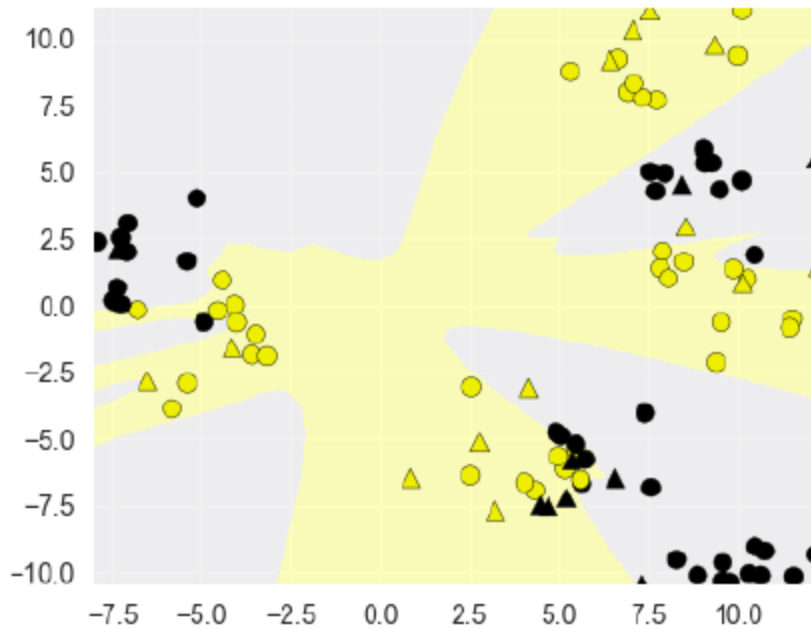


Dataset 1: Neural net classifier, 2 layers, 10/10 units
Train score = 0.93, Test score = 0.68

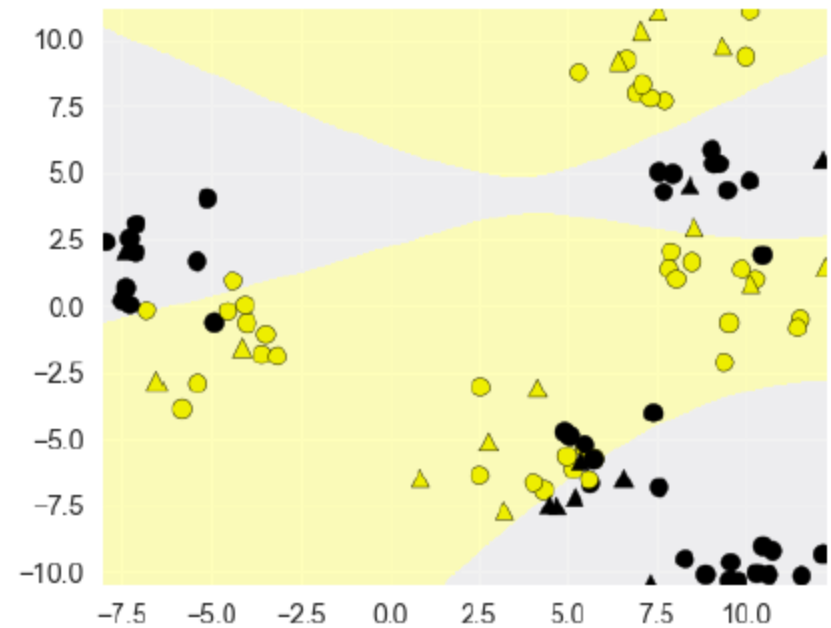


Efeito de alfa

Dataset 2: NN classifier, $\alpha = 0.010$
Train score = 0.97, Test score = 0.72



Dataset 2: NN classifier, $\alpha = 5.000$
Train score = 0.87, Test score = 0.92



MLP para regressão

- ▶ Semelhante a classificação

```
from sklearn.neural_network import MLPRegressor
mlpreg = MLPRegressor(hidden_layer_sizes = [100,100],
                      activation = 'relu',
                      alpha = 1,
                      solver = 'lbfgs').fit(X_train, y_train)
```



Rede MLP

- ▶ **Vantagens**

- ▶ Boa performance
- ▶ Formam a base para algoritmos estado da arte em muitas aplicações

- ▶ **Desvantagens**

- ▶ Grande tempo de treinamento
- ▶ Grande quantidade de hiperparâmetros



Aprendizado Não Supervisionado

Aprendizado Não Supervisionado

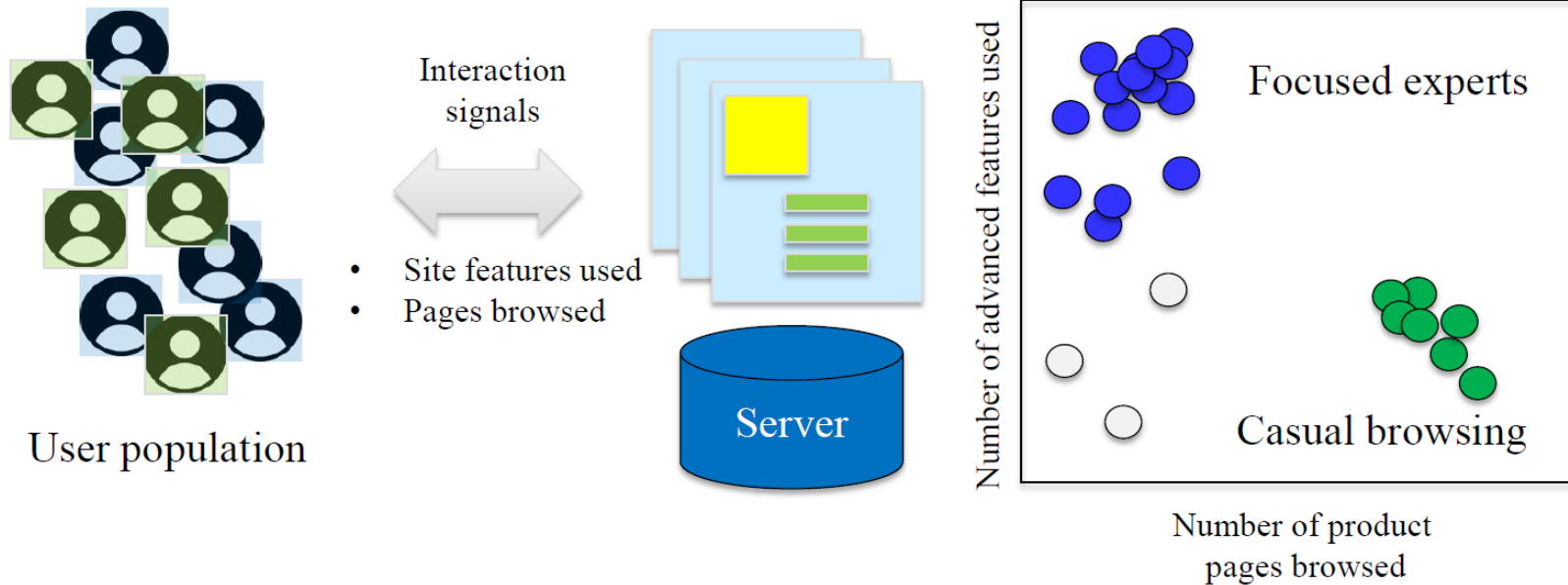
- ▶ Tarefa de analisar dados que não possuem rótulos
- ▶ Inferir a estrutura do conjunto de dados

- ▶ Tarefas não supervisionadas
 - ▶ Visualizar a estrutura de dados multidimensionais
 - ▶ Comprimir ou sumarizar dados
 - ▶ Descobrir agrupamentos (clusters) ou outliers



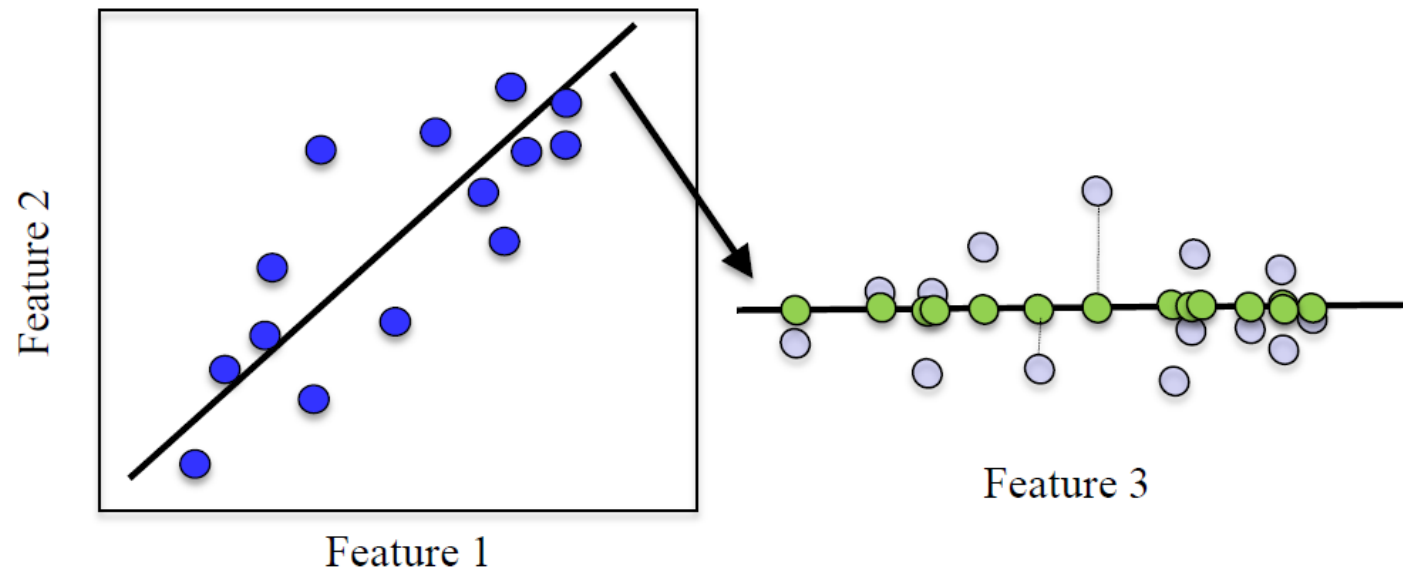
Agrupamento (clustering)

- ▶ Segmentação
- ▶ Tratamento diferenciado



Redução de dimensionalidade

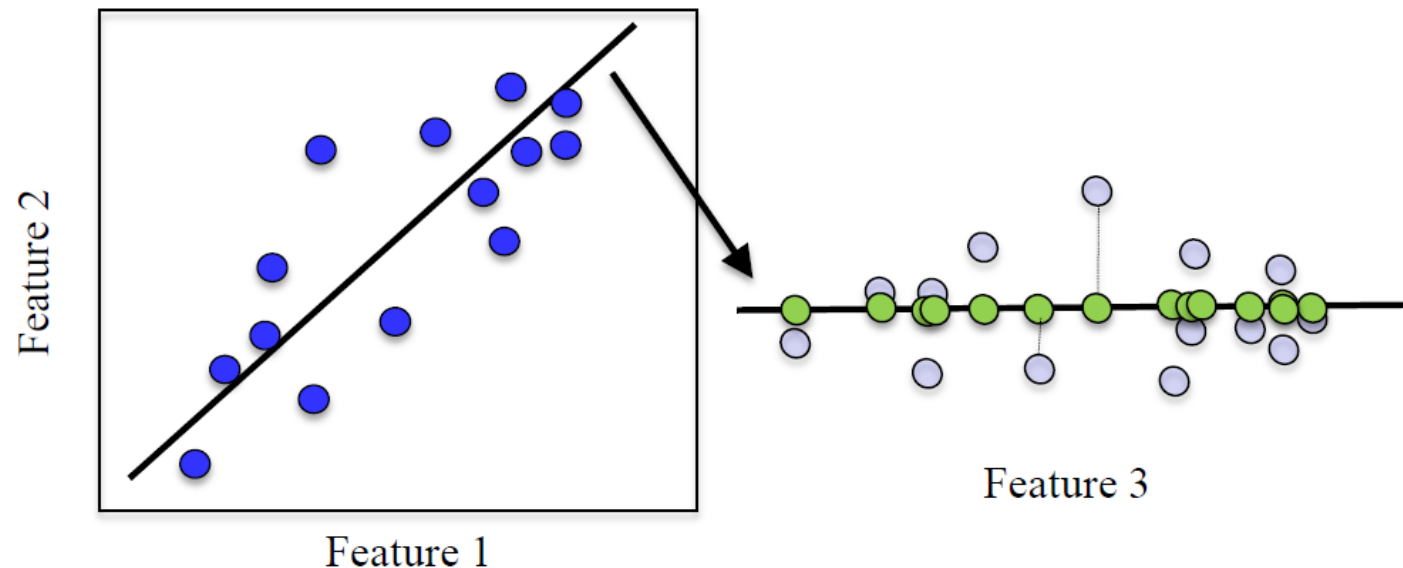
- ▶ Encontrar aproximação do conjunto de dados original em um espaço de dimensão reduzido



Redução de dimensionalidade

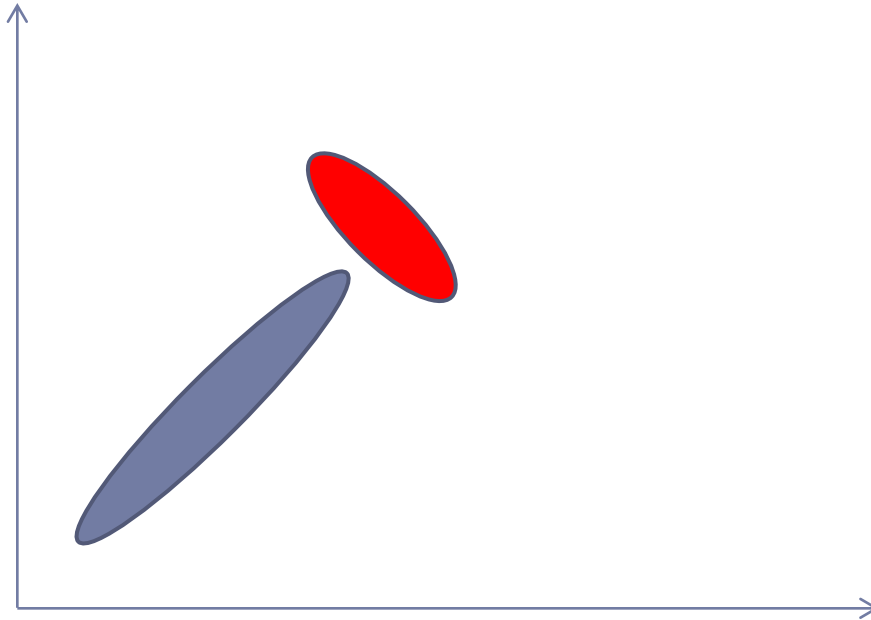
Redução de dimensionalidade

- ▶ Visualização dos dados
- ▶ Redução da complexidade de modelos de AM



Análise de Componentes Principais (PCA)

► Idéia



PCA

- ▶ Informação = Variância
- ▶ Rotacionar os dados
 - ▶ Transformação linear
 - ▶ Novos atributos são um combinação linear dos atributos originais



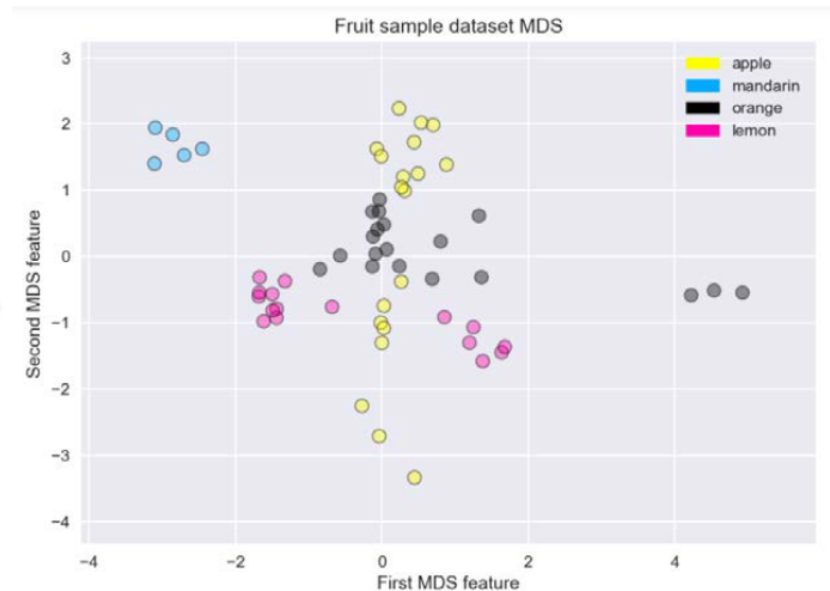
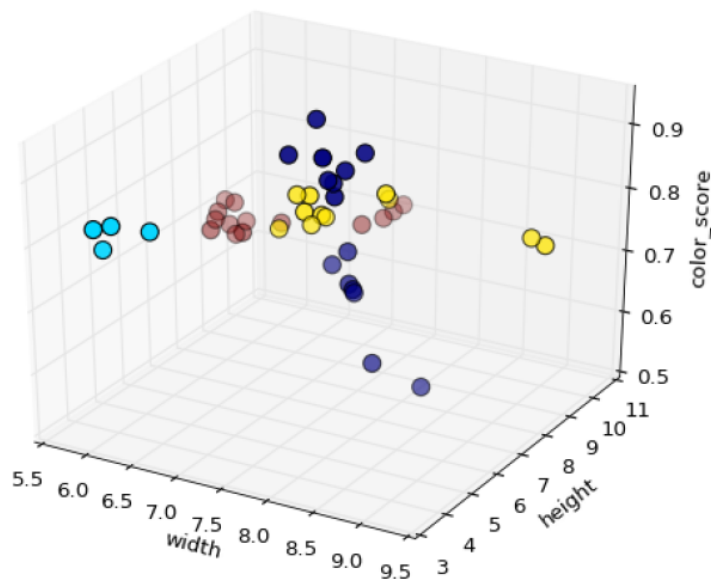
PCA na prática

- ▶ Toma os dados de entrada
- ▶ Subtrai as médias
- ▶ Calcula a matriz de covariância
- ▶ Calcula os autovalores e autovetores
- ▶ Escolhe os k maiores autovalores
- ▶ Utiliza os k autovetores correspondentes para criar k novos atributos



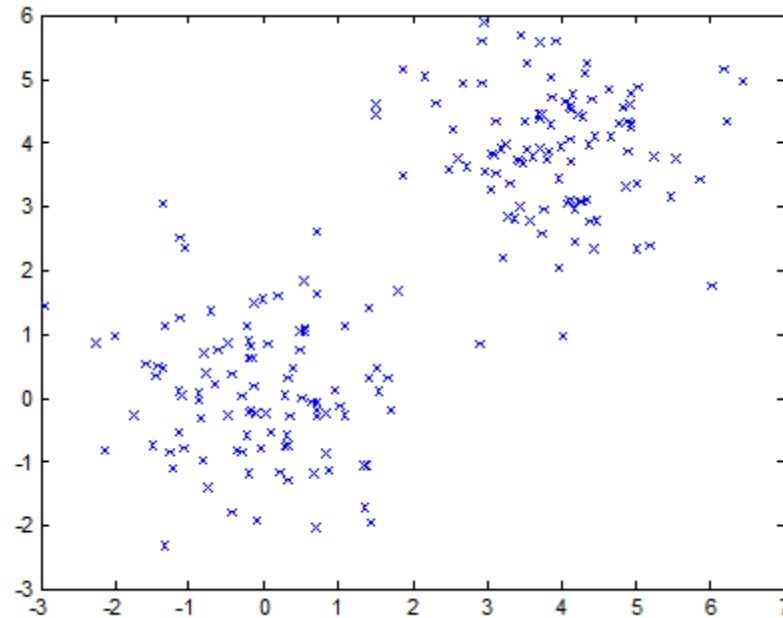
Escalonamento Muldimensional (MDA)

- ▶ Transformação nos dados que preserva relações de vizinhança
- ▶ Somente visualização

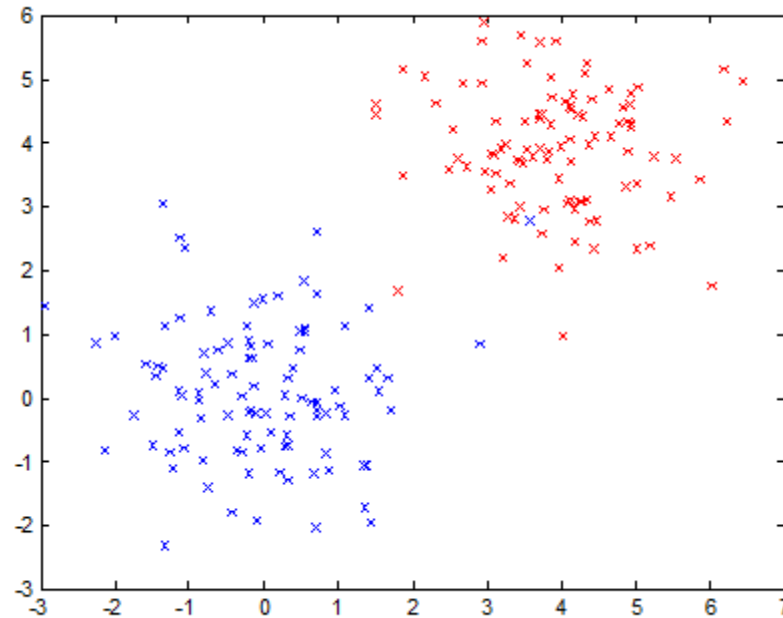


Clustering

Classificação Não Supervisionada



Classificação Não Supervisionada



Classificação Não Supervisionada

- ▶ Quantos grupos existem ?
- ▶ Quais os componentes destes grupos ?



Classificação Não Supervisionada

- ▶ Quantos grupos existem ?
- ▶ Quais os componentes destes grupos ?
- ▶ Métodos de Agrupamento
 - ▶ Hierárquico
 - ▶ Não Hierárquico



Agrupamento Hierárquico

- ▶ Os dados iniciam em grupos definidos
- ▶ Dados similares são agrupados formando pequenos grupos
- ▶ Pequenos grupos são agrupados formando grupos maiores
- ▶ Procedimento é repetido até que todos pertençam a um grupo



Agrupamento Hierárquico

- ▶ Dados são ditos semelhantes de acordo com alguma medida de distância.



Agrupamento Hierárquico

- ▶ Dados são ditos semelhantes de acordo com alguma medida de distância.

- ▶ Euclidiana

- ▶ $D_E(\mathbf{r}, \mathbf{s}) = \sqrt{\sum_{j=1}^p (r_j - s_j)^2}$

- ▶ Manhattan

- ▶ $D_M(\mathbf{r}, \mathbf{s}) = \sum_{j=1}^p |r_j - s_j|$



Agrupamento Hierárquico

- ▶ Dados são ditos semelhantes de acordo com alguma medida de distância.
 - ▶ Euclidiana
 - ▶ $D_E(\mathbf{r}, \mathbf{s}) = \sqrt{\sum_{j=1}^p (r_j - s_j)^2}$
 - ▶ Manhattan
 - ▶ $D_M(\mathbf{r}, \mathbf{s}) = \sum_{j=1}^p |r_j - s_j|$
- ▶ Similaridade entre grupos pode ser medida pela distância entre centróides
- ▶ Gráfico semelhante a uma árvore



Agrupamento Hierárquico

- ▶ Exemplo

- ▶ Dados

- ▶ [1 2],[1 1],[3 3] e [4 3]

- ▶ Calcula-se uma matriz de distâncias (d^2)

	1	2	3	4
1	0	1	5	10
2	1	0	8	13
3	5	8	0	1
4	10	13	1	0



Agrupamento Hierárquico

► [1 2],[1 1],[3 3] e [4 3]

	1	2	3	4
1	0	1	5	10
2	1	0	8	13
3	5	8	0	1
4	10	13	1	0

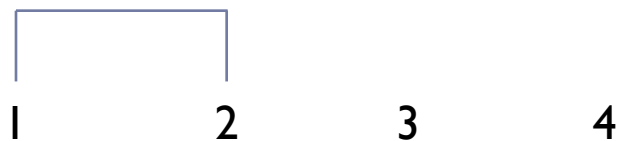


Agrupamento Hierárquico

► [1 2],[1 1],[3 3] e [4 3]

	1,2	3	4
1,2	0		
3		0	1
4		1	0

$$c_{1,2} = [1 \ 1,5]$$



Agrupamento Hierárquico

► [1 1.5],[3 3] e [4 3]

	1,2	3	4
1,2	0	6.25	11.25
3	6.25	0	1
4	11.25	1	0

$$c_{1,2} = [1 \ 1,5]$$



Agrupamento Hierárquico

► $[1 \ 1.5], [3 \ 3]$ e $[4 \ 3]$

	1,2	3	4
1,2	0	6.25	11.25
3	6.25	0	1
4	11.25	1	0

$$c_{1,2} = [1 \ 1.5]$$



Agrupamento Hierárquico

► [1 1.5],[3 3] e [4 3]

	1,2	3,4
1,2	0	
3,4		0

$$C_{3,4} = [3.5 \ 3]$$



Agrupamento Hierárquico

► [1 1.5] e [3.5 3]

	1,2	3,4
1,2	0	8
3,4	8	0

$$C_{3,4} = [3.5 \ 3]$$

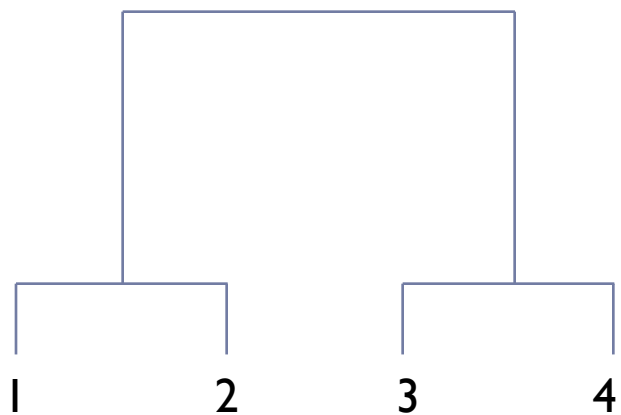


Agrupamento Hierárquico

► [1 1.5] e [3.5 3]

	1,2	3,4
1,2	0	8
3,4	8	0

$$C_{3,4} = [3.5 \ 3]$$

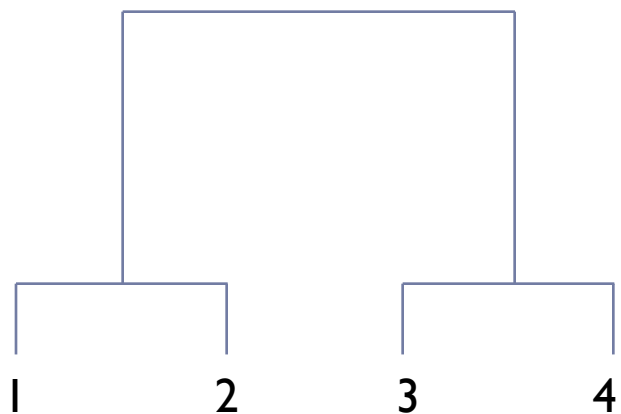


Agrupamento Hierárquico

► [1 1.5] e [3.5 3]

	1,2	3,4
1,2	0	8
3,4	8	0

$$C_{3,4} = [3.5 \ 3]$$



Dendrograma

Agrupamento não Hierárquico

- ▶ Pontos pertencem a algum grupo
- ▶ Pontos mudam de grupo de forma a satisfazer um determinado critério



K-médias

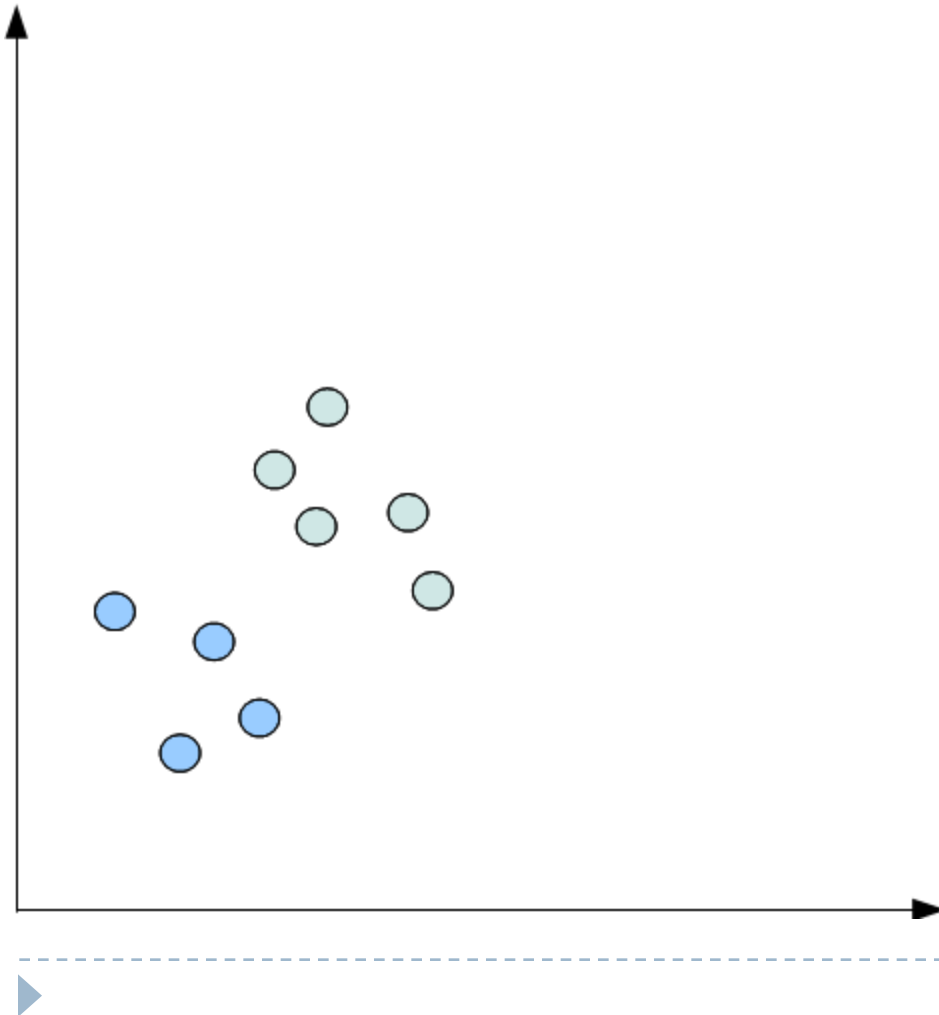


K-médias

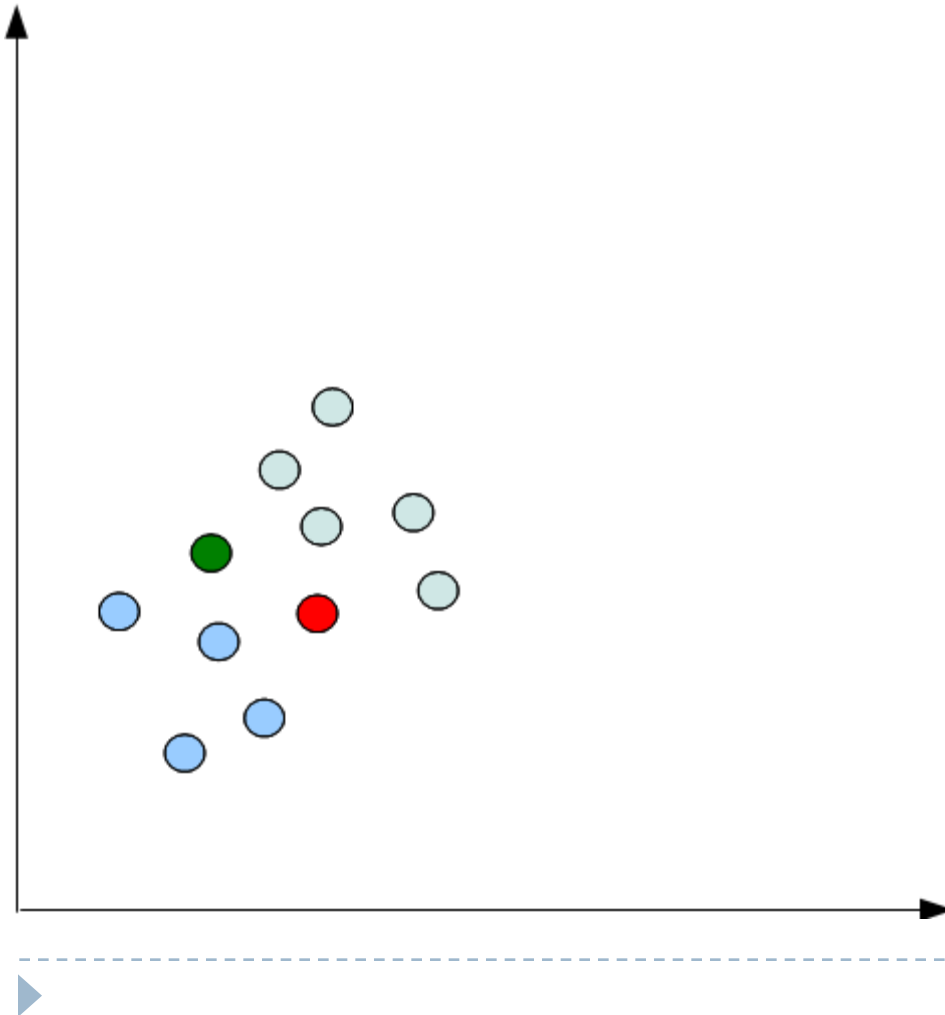
- ▶ É necessário conhecer o número de clusters
- ▶ k centroides são escolhidos aleatoriamente (podem ser escolhidos k membros da população)
- ▶ Calcula-se a distância destes pontos para todos os outros
- ▶ Os pontos passarão a pertencer ao grupo cuja distância é a menor
- ▶ Centróides são recalculados como a média dos pontos do grupo



K-médias

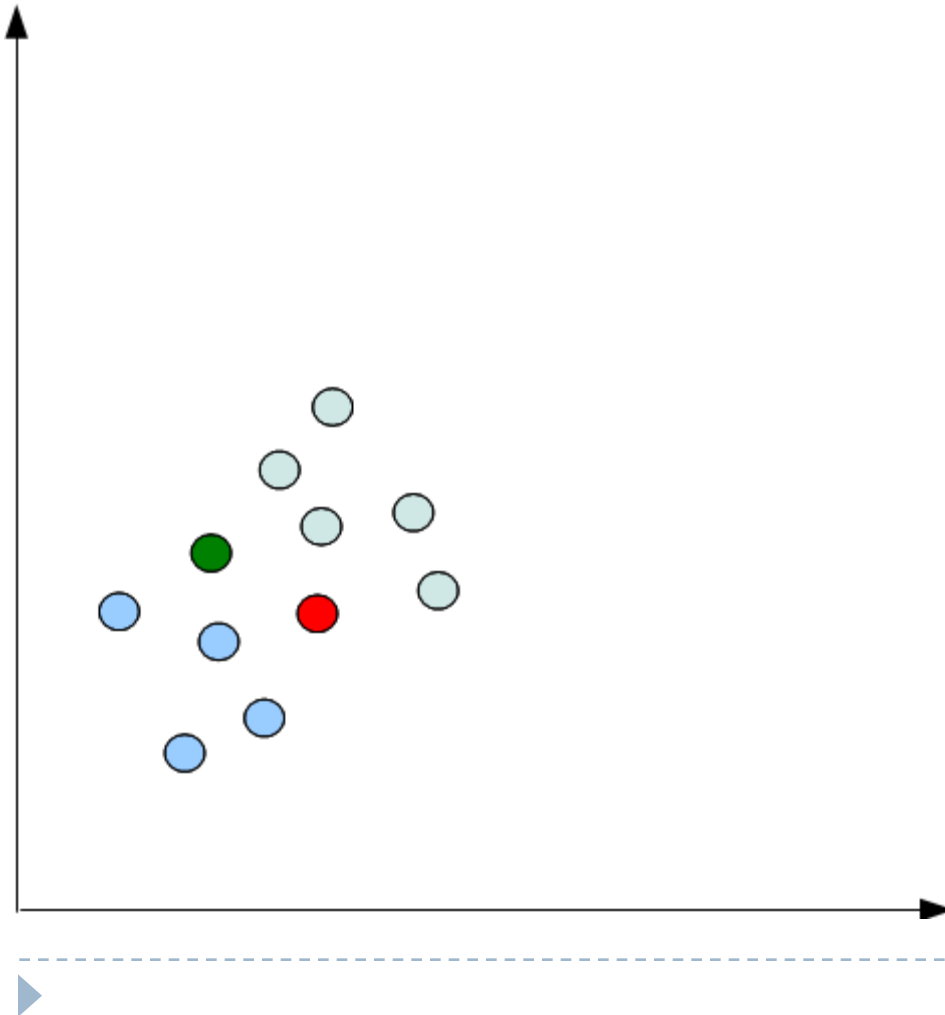


K-médias



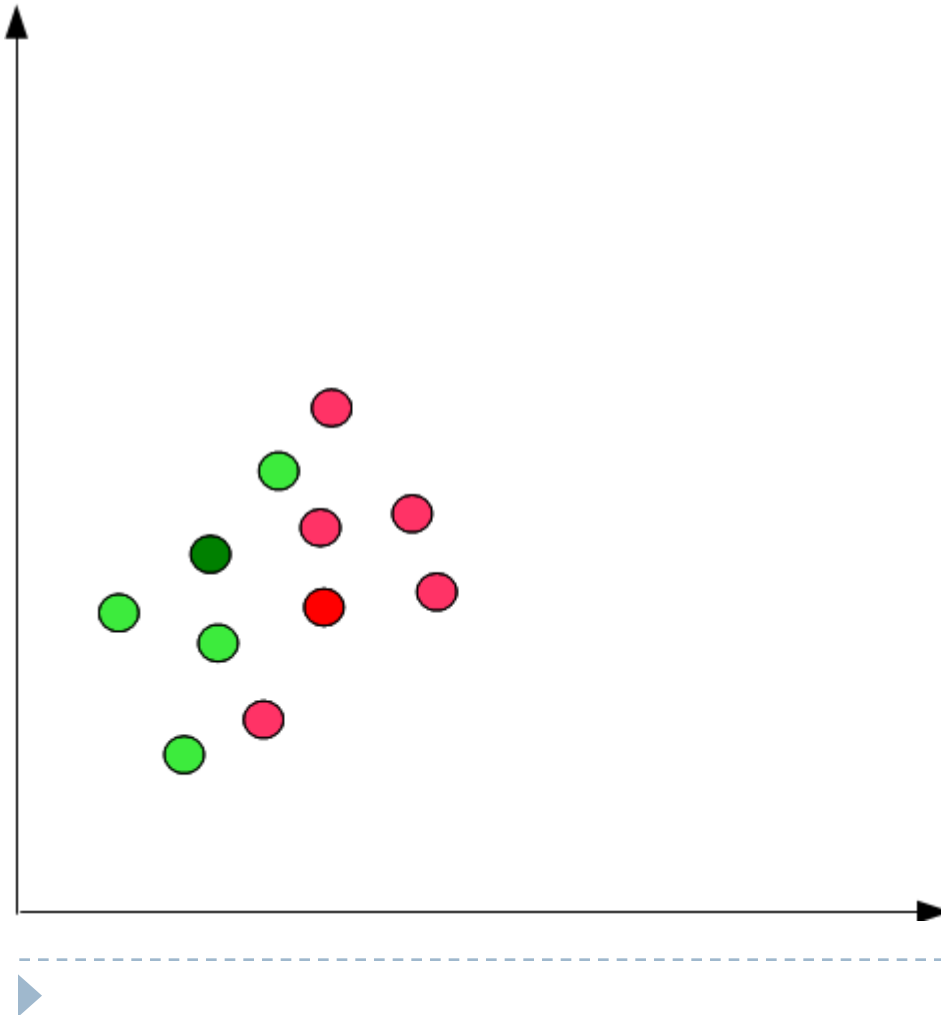
K-médias

I – Calcular distâncias



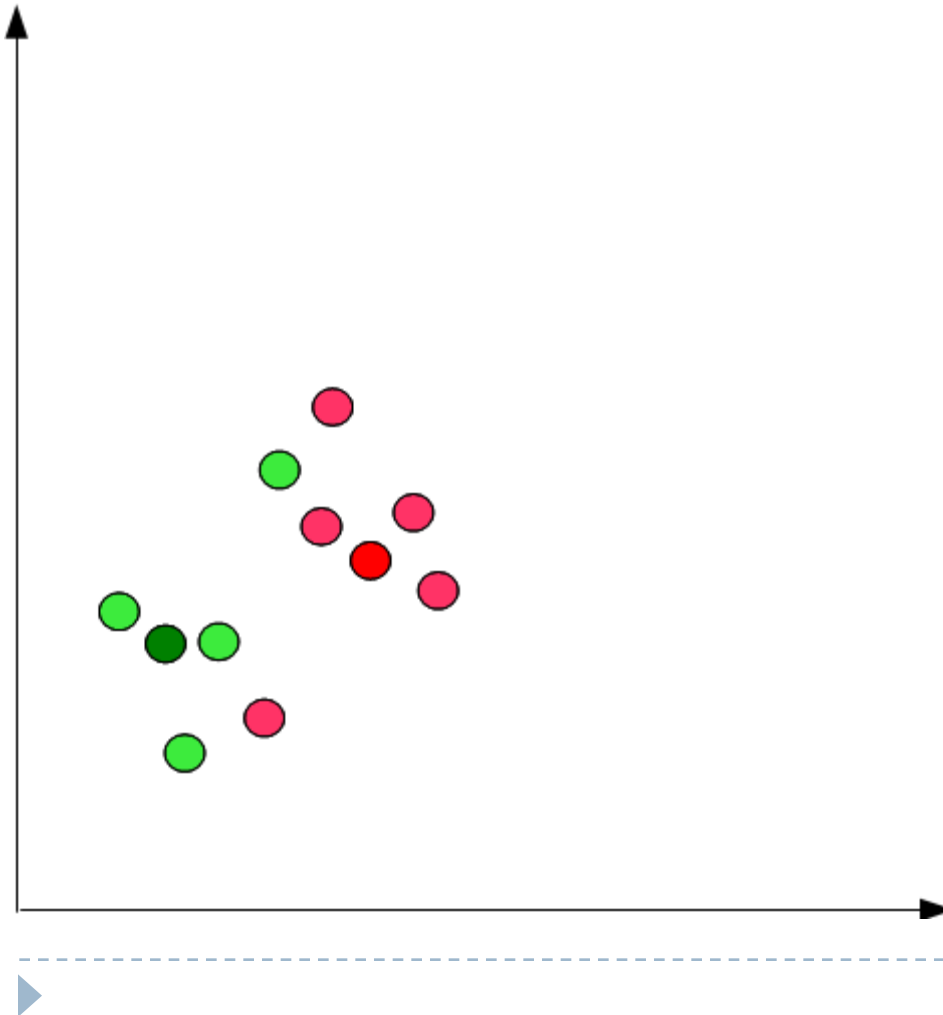
K-médias

- 1 – Calcular distâncias
- 2 – Atribuir Grupos



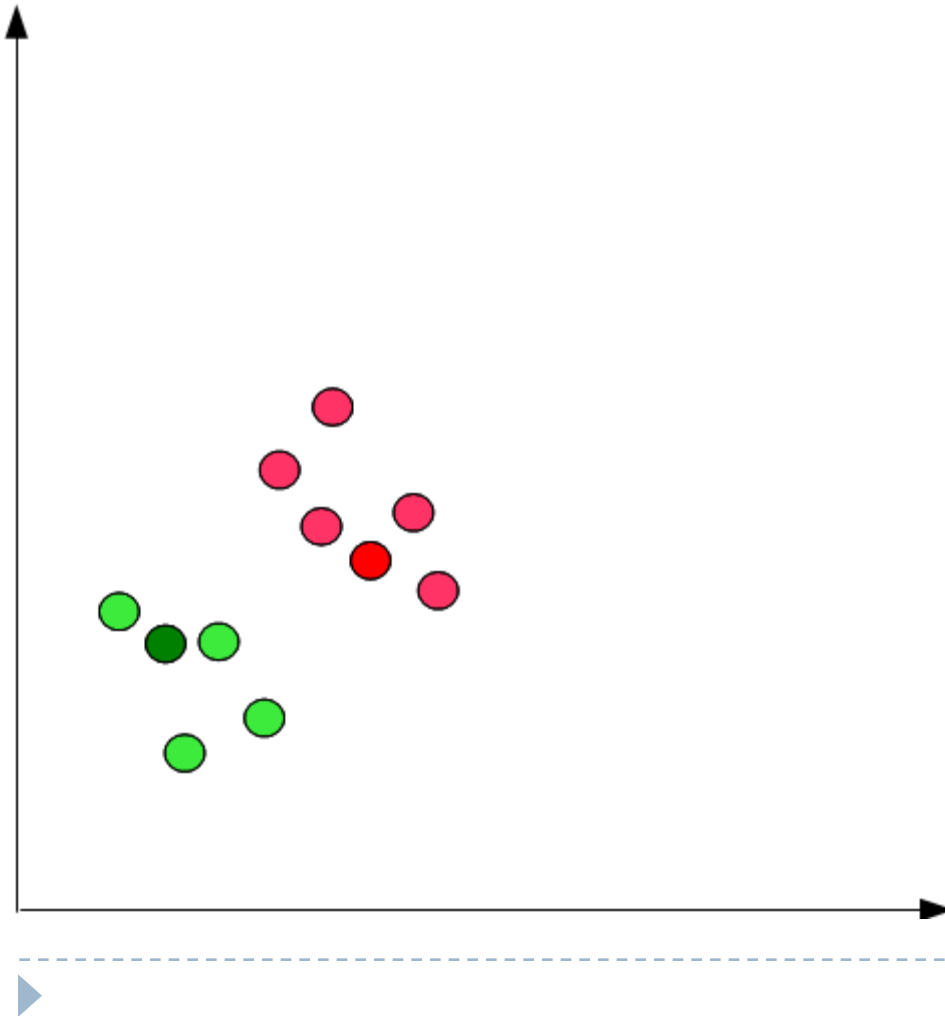
K-médias

- 1 – Calcular distâncias
- 2 – Atribuir Grupos
- 3 – Recalcular centróides



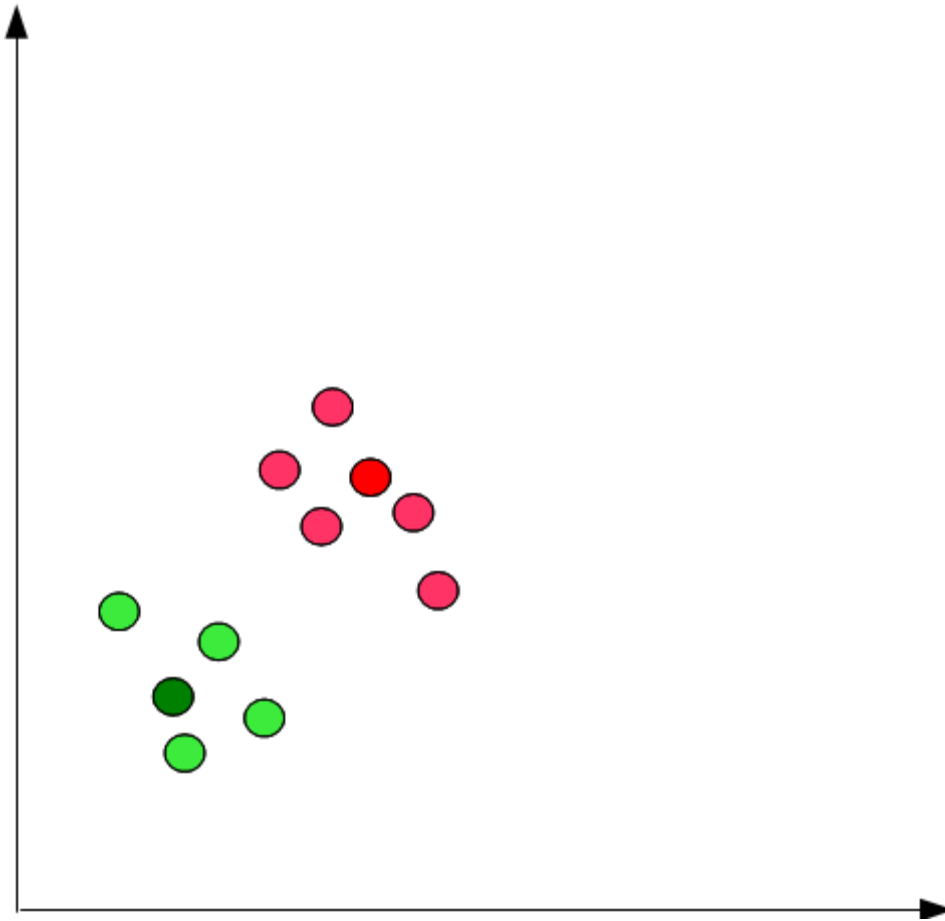
K-médias

- 1 – Calcular distâncias
- 2 – Atribuir Grupos
- 3 – Recalcular centróides



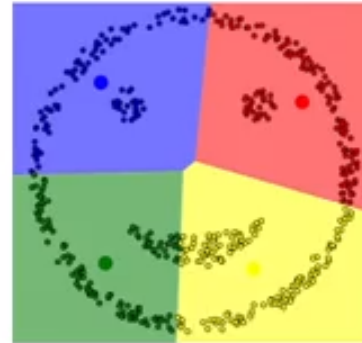
K-médias

- 1 – Calcular distâncias
- 2 – Atribuir Grupos
- 3 – Recalcular centróides



Limitações do K-médias

- ▶ Funciona bem para clusters de tamanhos parecidos, bem separados e com aspecto de hyper-esfera
- ▶ Atributos numéricos

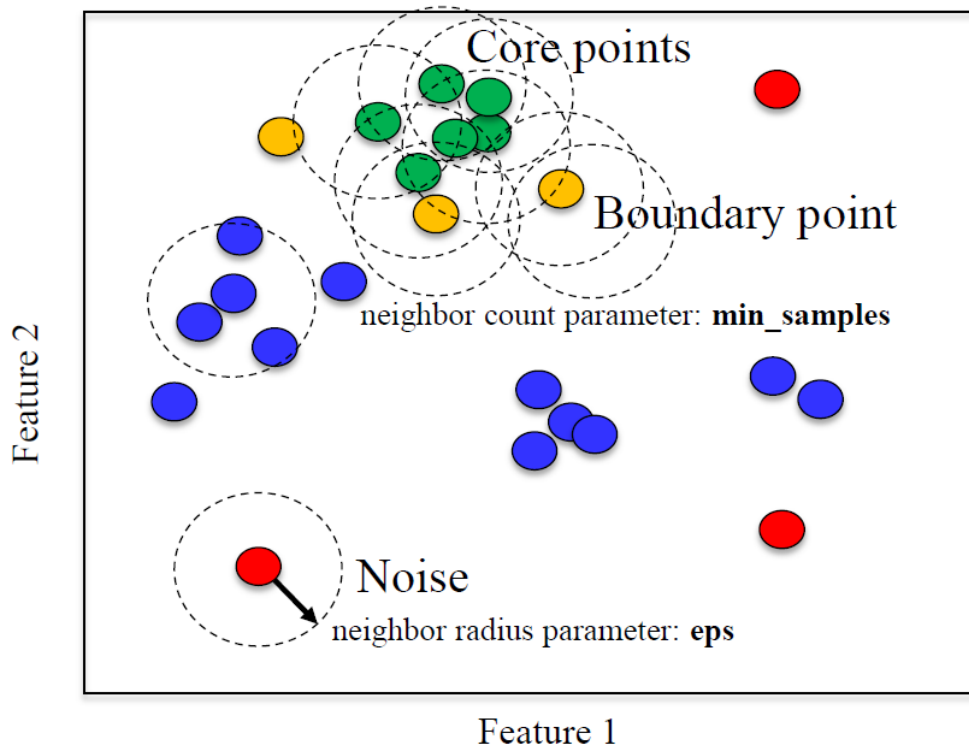


DBSCAN

- ▶ **Density based spatial clustering for applications with noise**
 - ▶ Não precisa especificar o numero de classes
 - ▶ Clusters mais complexos
 - ▶ Identifica outliers automaticamente
- ▶ **Idéia**
 - ▶ Clusters são áreas onde existem muitos pontos separados por regiões vazias
- ▶ **Parâmetros**
 - ▶ min_samples, eps



DBSCAN



- ▶ Core Points – Ponto que possui pelo menos **min_samples** dentro da região definida por **eps**
- ▶ Todos os Core points que estão a uma distancia **eps**, são colocados no mesmo cluster
- ▶ Pontos que não pertencem a um grupo são outliers
- ▶ Pontos a uma distancia **eps** de Core points que não são core points, serão boundary points