Aula 9 - Regressão

1001524 – Aprendizado de Máquina I
2023/1 - Turmas A, B e C

Prof. Dr. Murilo Naldi

naldi@ufscar.br

Agradecimentos

- Parte do material utilizado nesta aula foi disponibilizado por M. Kumar no endereço:
  - www-users.cs.umn.edu/~kumar/dmbook/index.php
- Agradecimentos a Intel Software e a

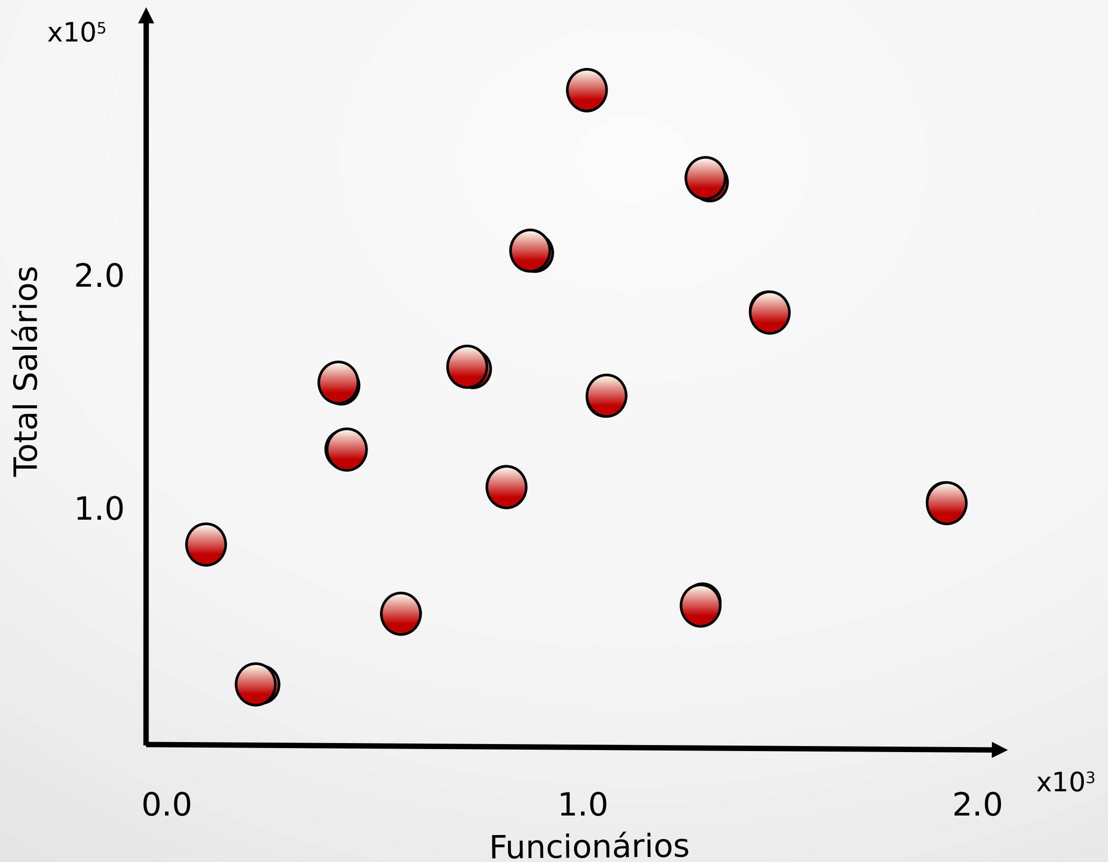
  Intel IA Academy pelo material disponibilizado e recursos didáticos

Regressão

- Regressão é uma técnica que permite inferir a relação de uma variável dependente (atributo objetivo) com variáveis independentes específicas (outros atributos).
- Usada como um método descritivo da análise de dados
- Designa uma equação matemática que descreva a relação entre duas ou mais variáveis.
  - Quando aplicado a mineração, essa equação é um modelo que representa os dados

Regressão Linear

- Considere o conjunto dados sobre folha de pagamento de algumas empresas



Regressão Linear

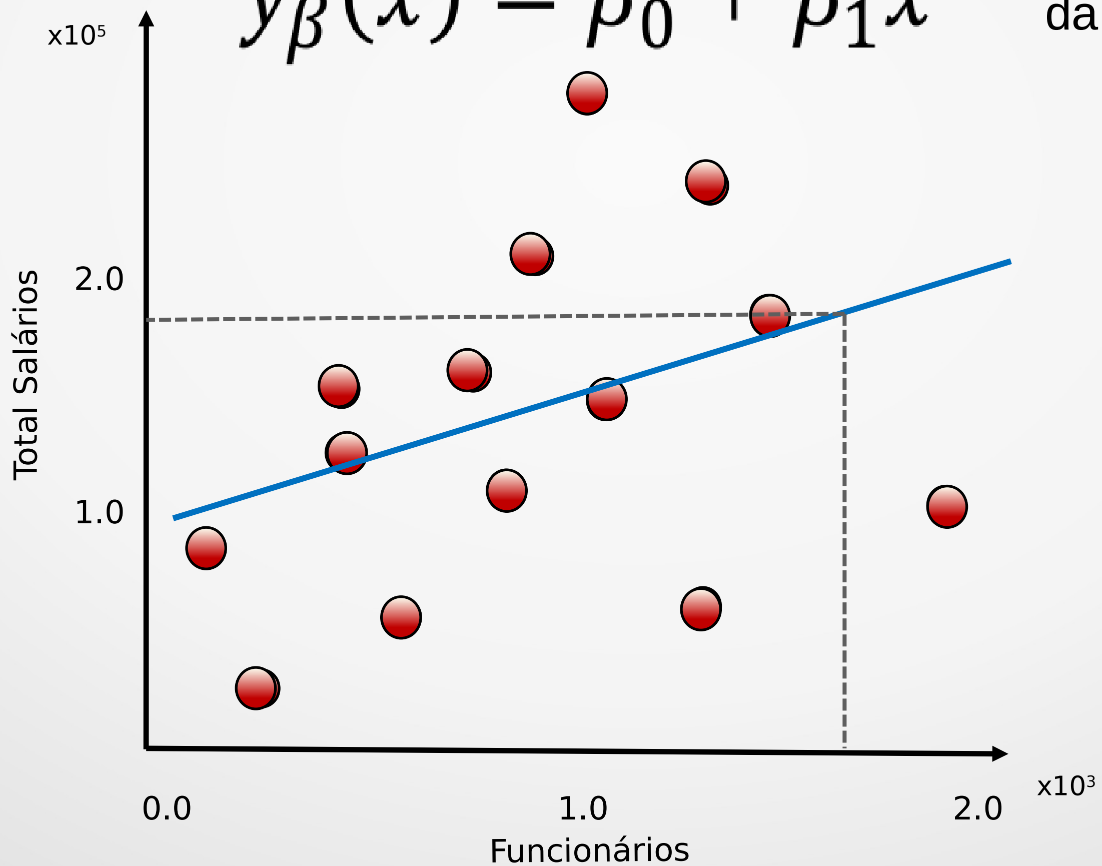
Total da folha de pagamento

Interceptação da reta com o eixo vertical

Coeficiente angular da reta

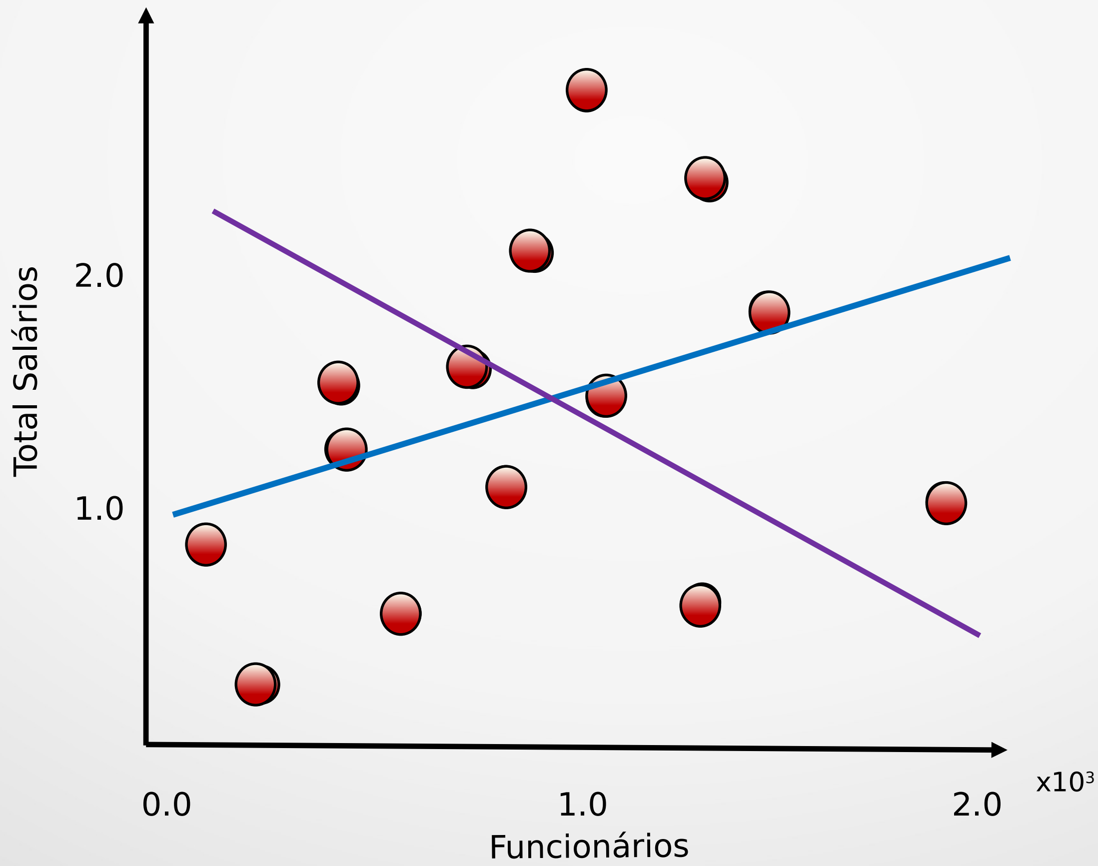
Número de funcionários da empresa

$$y_\beta(x) = \beta_0 + \beta_1 x$$



Regressão Linear

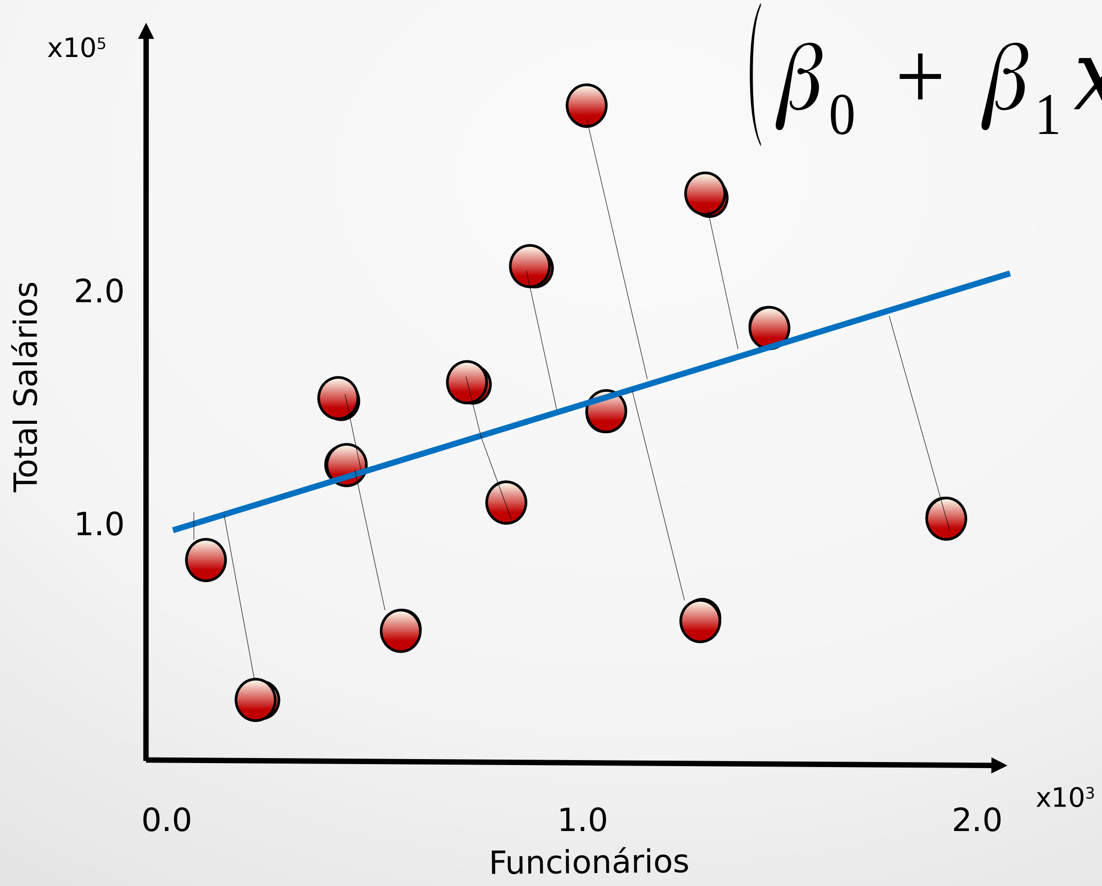
- Mas qual modelo escolher? Afinal, existem diversos!



Regressão Linear

- Cada modelo possui um erro que é a diferença entre o valor predito e o conhecido

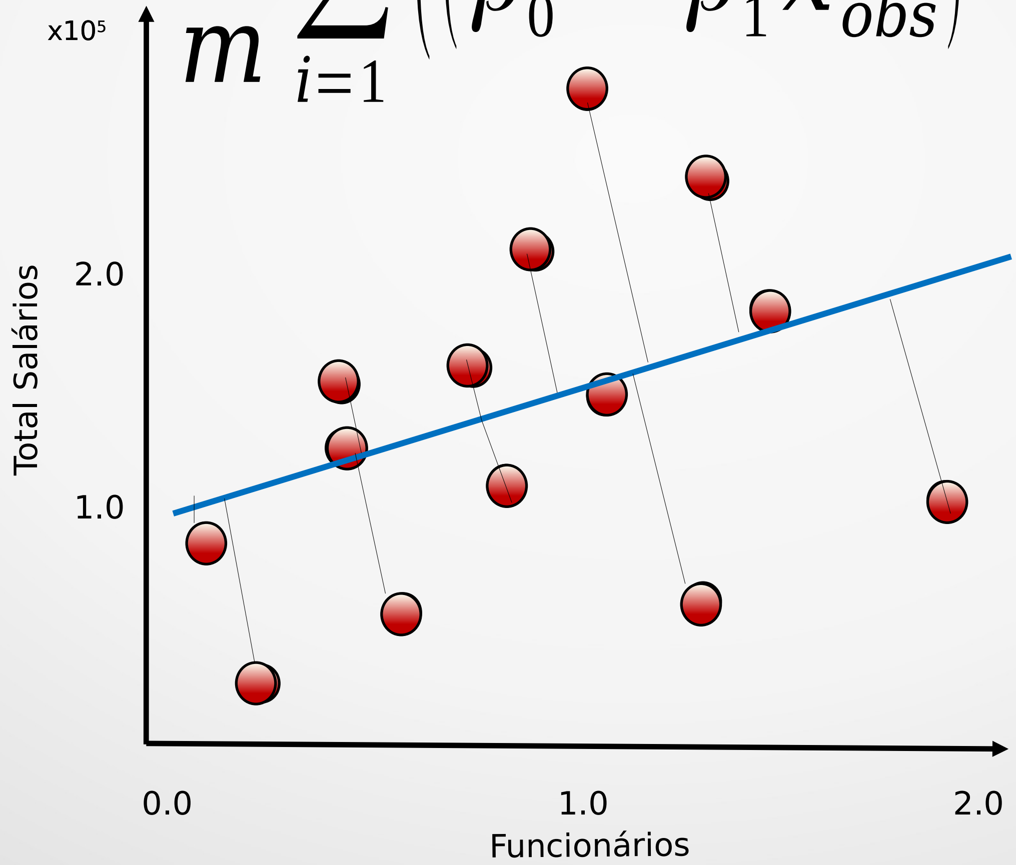
$$\left(\beta_0 + \beta_1 x_{obs}^{(i)}\right) - y_{obs}^{(i)}$$



Calculando Erro

- Erro médio quadrático (Mean Squared Error - MSE)

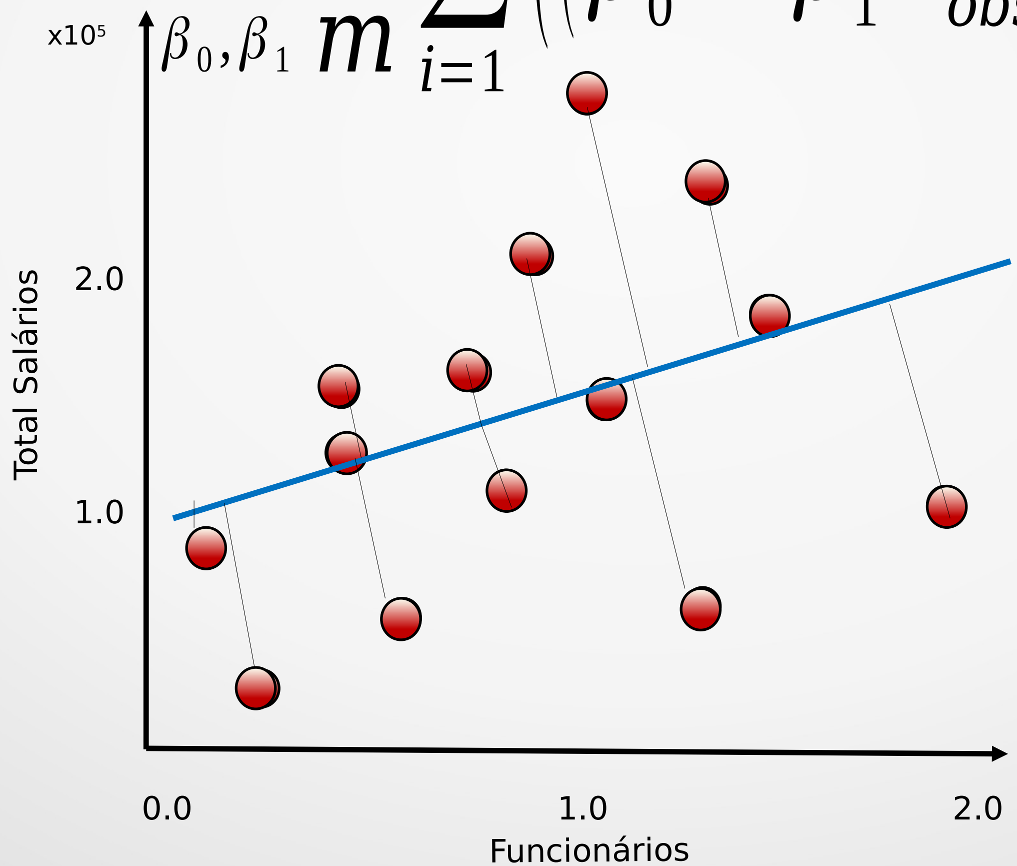
$$\frac{1}{m}\sum_{i=1}^{m}\left(\left(\beta_0 + \beta_1 x_{obs}^{(i)}\right) - y_{obs}^{(i)}\right)^2$$



Escolhendo Modelo

- Gerar diferentes modelos e escolher o menor MMSE

$$\underset{\beta_0,\beta_1}{m \in} \frac{1}{m}\sum_{i=1}^{m}\left(\left(\beta_0 + \beta_1 x_{obs}^{(i)}\right) - y_{obs}^{(i)}\right)^2$$

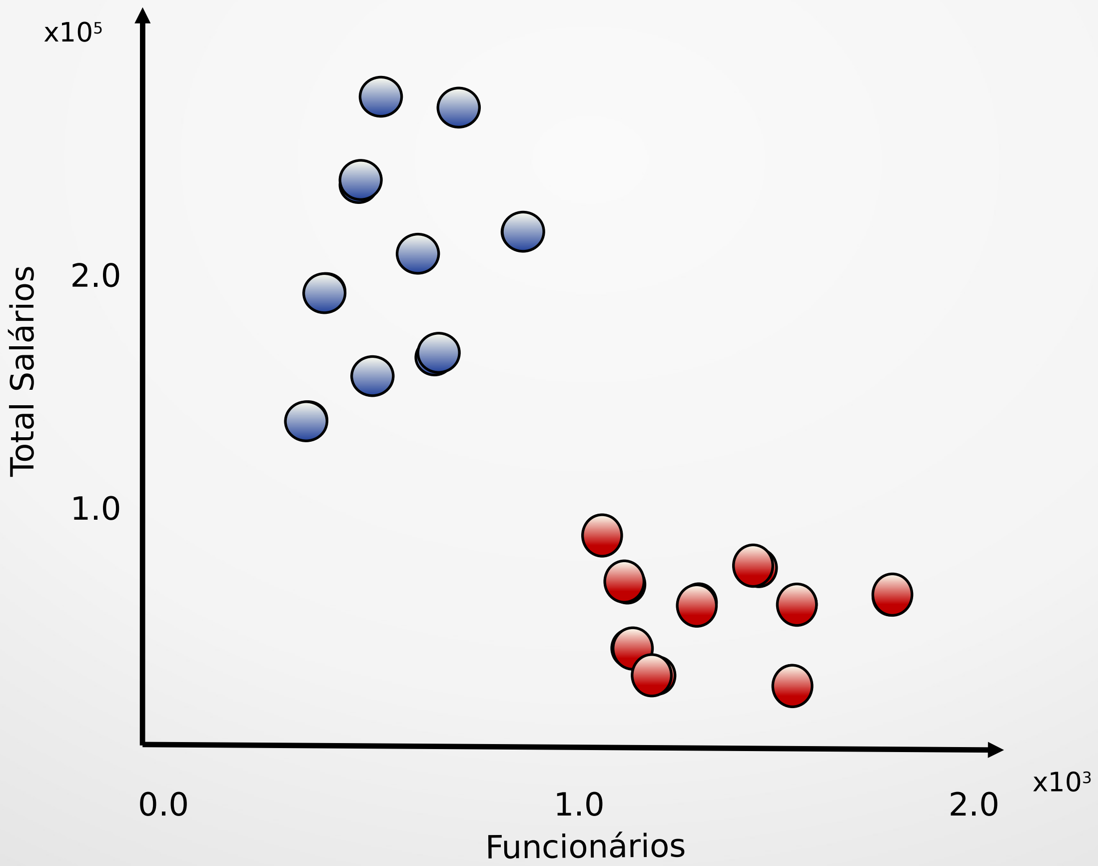


Ajustando a regressão linear

- O ajuste aos dados é feito de forma a encontrar valores para β_0 e β_1 de forma a minimizar o MSE
  - O que pode ser feito utilizando por um processo de otimização
    - Por exemplo, usando derivadas parciais ou transformação de coordenadas
- A regressão pode ser feita para todo o conjunto de dados ou para parte dele

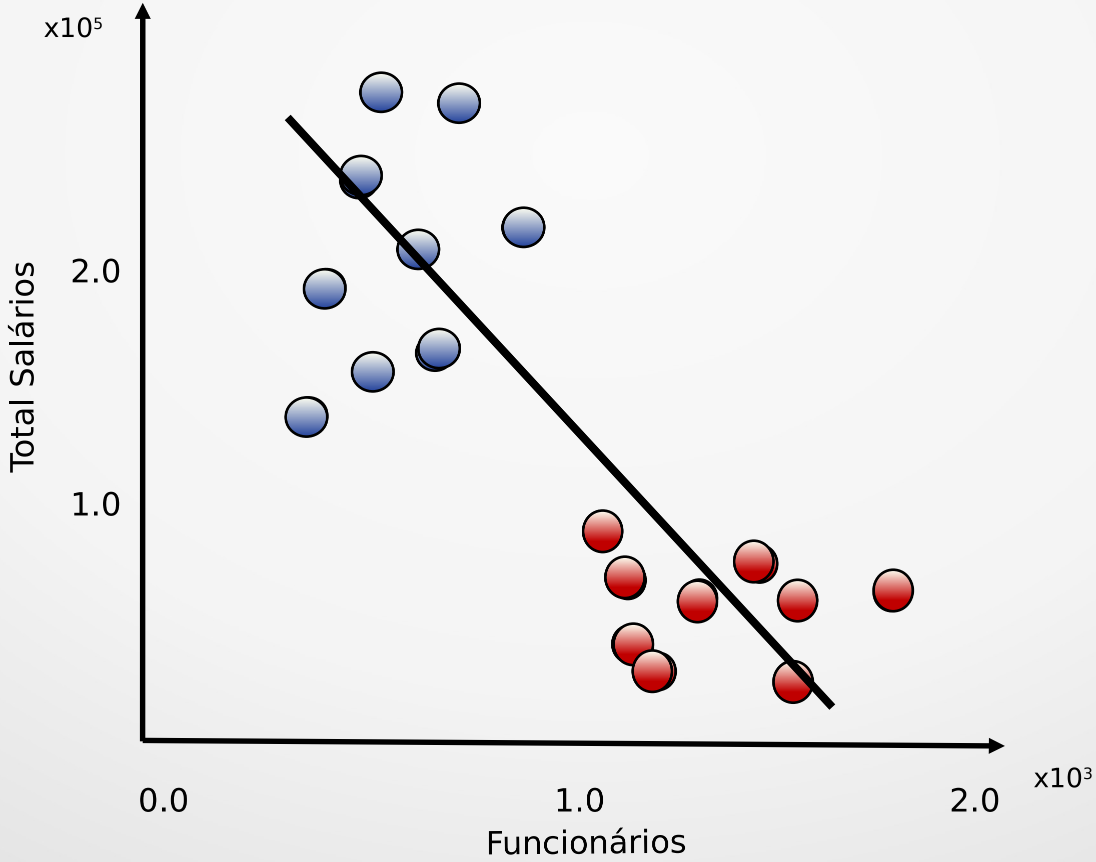
Ajuste do modelo

- No exemplo abaixo, colocamos empresas nacionais e multinacionais em cores distintas



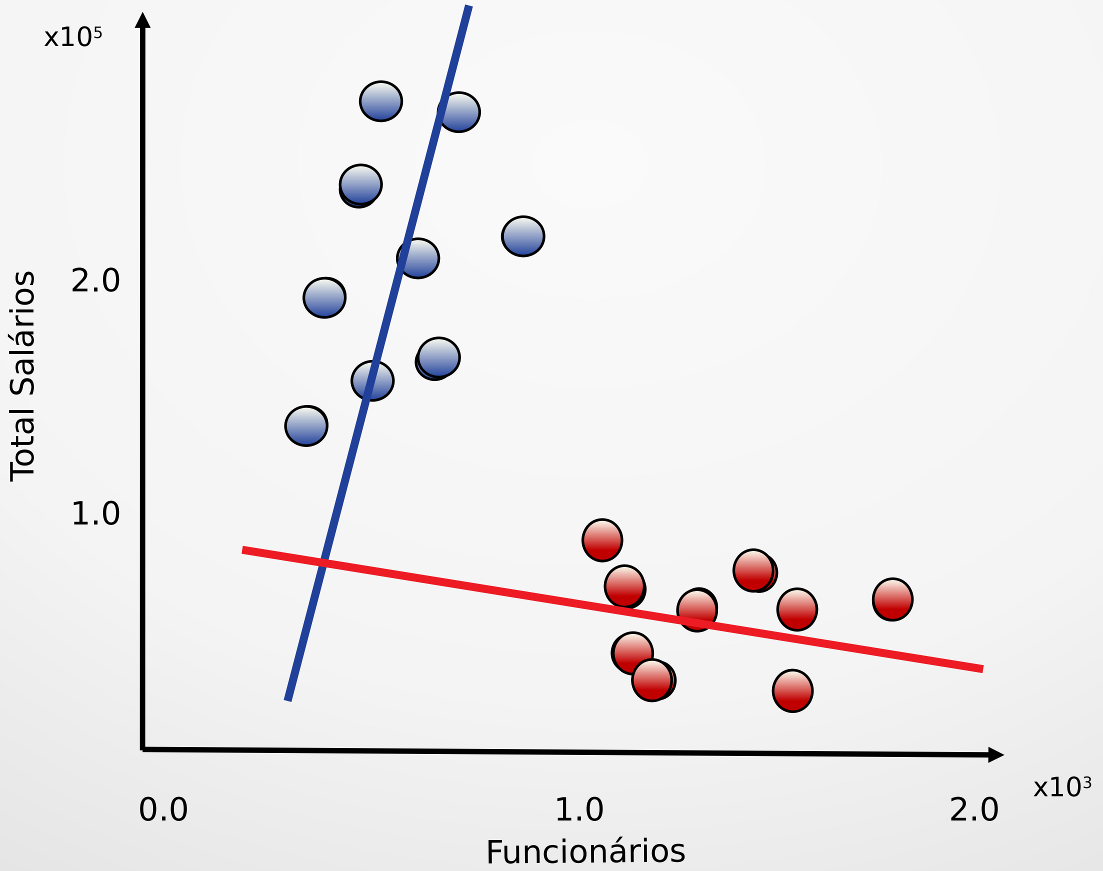
Ajuste do modelo

- Considerando o ajuste de todos os dados em uma única regressão



Ajuste do modelo

- Regressões distintas para dados de origens distintas podem possuir melhor ajuste



Aplicação

- Concrete Slump Test Data Set (UCI)
  - Mede o fluxo de queda de concreto
  - Importante para medir o quanto o concreto está maleável e, portanto, trabalhável
  - Ideia é gerar um modelo que possa predizer o fluxo de queda a partir dos exemplos do conjunto

Código Regressão Linear

```
import pandas as pd
#Importa o método de regressão do Sklearn
from sklearn.linear_model import LinearRegression
# Localização do arquivo
filepath = 'data/slump_test.csv'
# Importando os dados
data = pd.read_csv(filepath)
# Apagando a primeira coluna com ID
data = data.drop(columns=['No'])
#Colocando os dados em ordem aleatória
randomdata = (data.sample(n=103, replace=False))
#Aplicando hold out
traindata = randomdata.iloc[:85,:]
testdata = randomdata.iloc[85:,:]
#Cria uma instância da classe
LR = LinearRegression()
#Faz o ajuste da classe aos dados de cimento
LR = LR.fit(traindata.iloc[:,0:9], traindata.iloc[:,9])
#Classe predita
print(LR.predict(testdata.iloc[:,0:9]))
```

- Saída

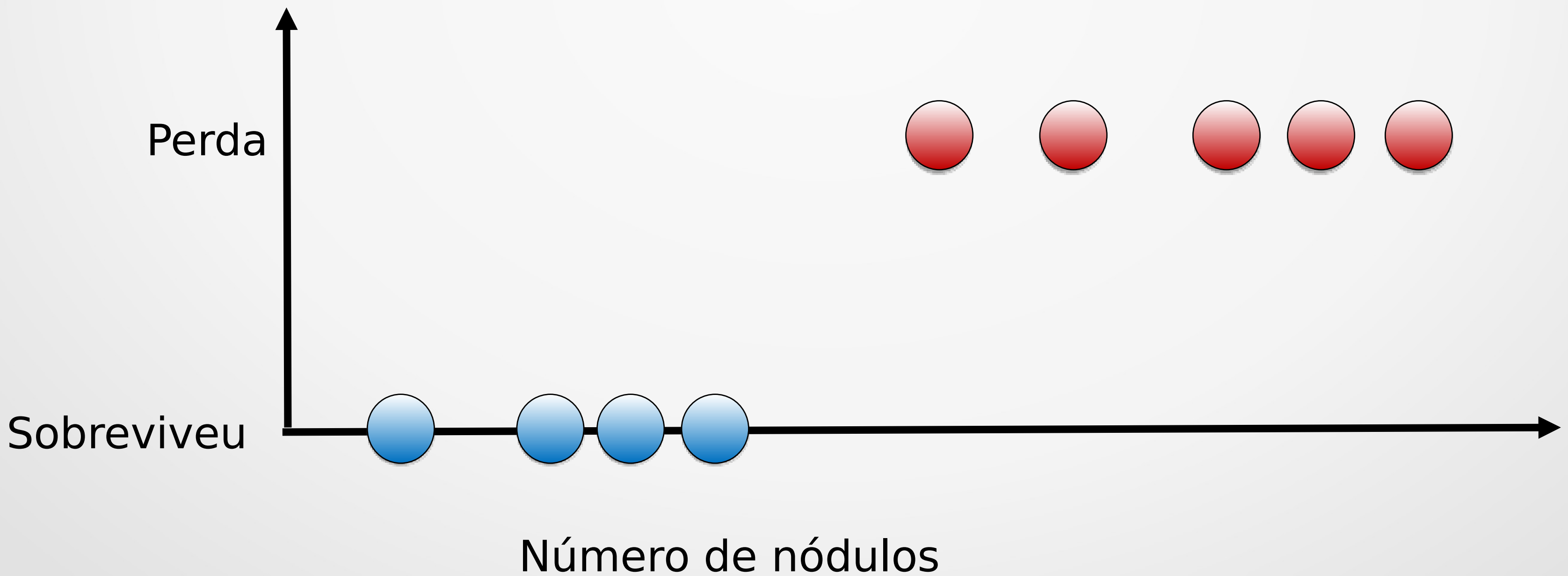
```
[49.04026231 31.23182173
33.0793428  35.09510413
46.30841661 43.46326958

 35.84775357 29.32540408
36.92580888 16.73161505
29.54350147 34.20239399

 28.13314981 38.95513352
27.00233273 39.73375906
36.43069133 39.41580758]
```

Considere o seguinte caso

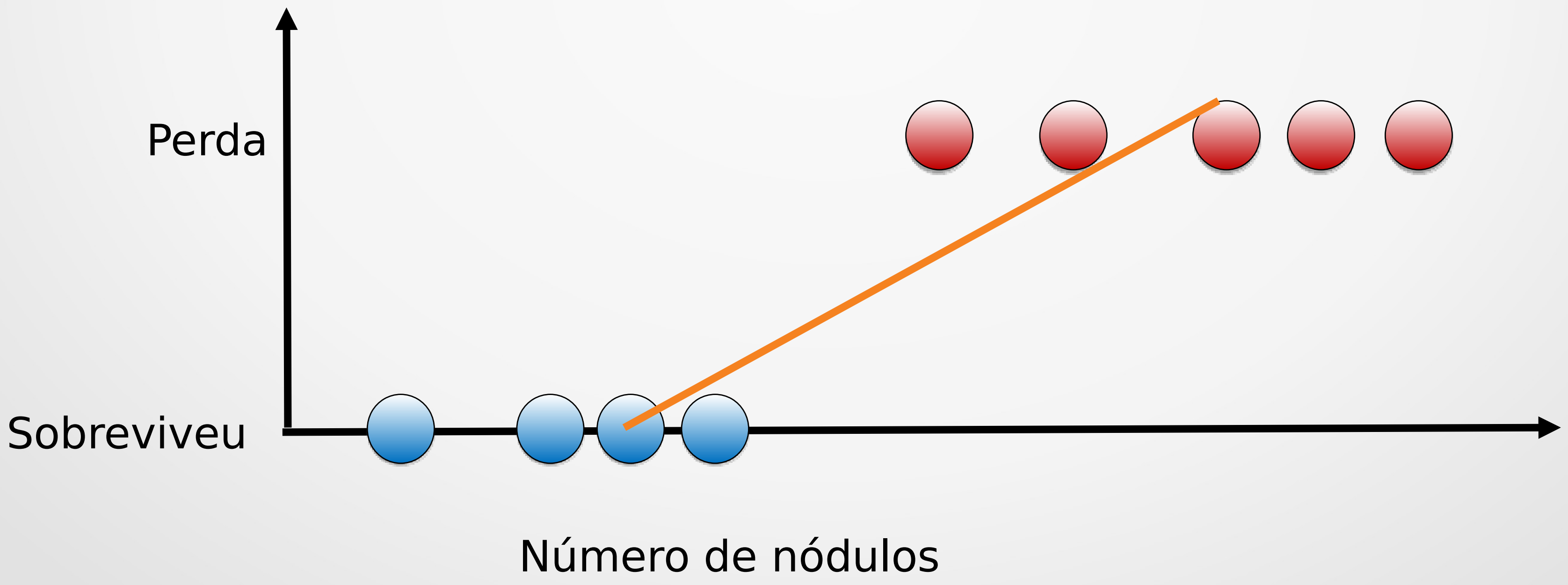
- Imagine que os dados a seguir são casos de câncer em que alguns pacientes faleceram e outros não em 5 anos



Usando regressão linear

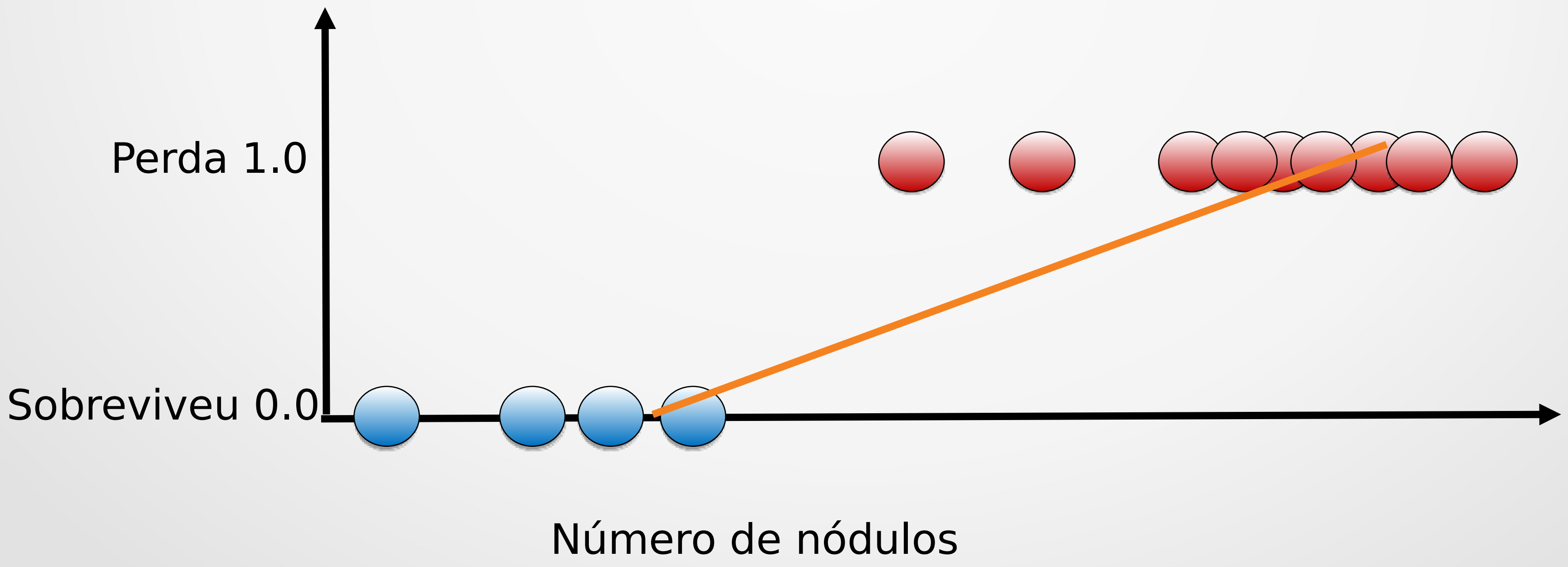
- Aplicando regressão linear sobre os dados, teremos o seguinte resultado

$$y_\beta(x) = \beta_0 + \beta_1 x + \varepsilon$$



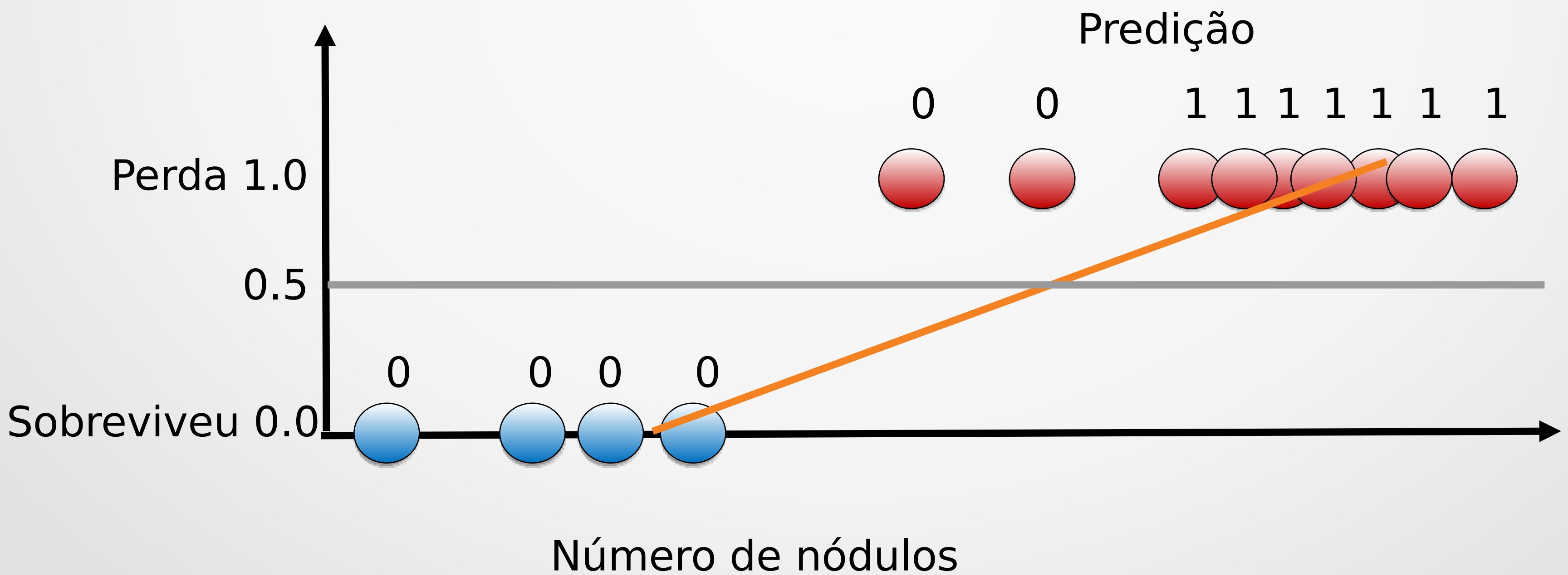
Usando regressão linear

- Suponhamos outro modelo, com mais dados em uma das classes de forma a influenciar mais a tendência da regressão



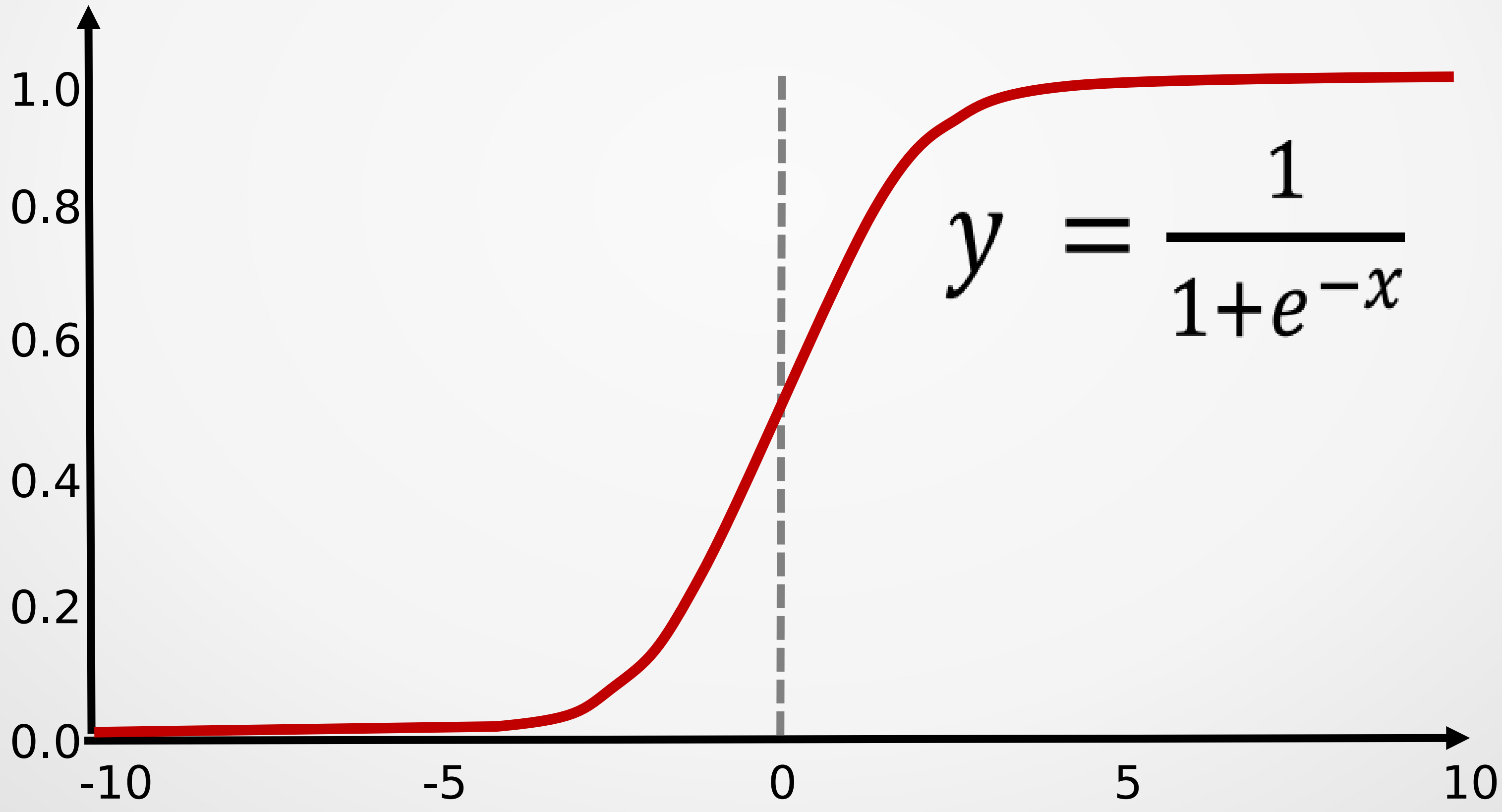
Usando regressão linear

- Usando o seguinte classificador:
  - Resultado do modelo > 0.5 = Perda
  - Resultado do modelo < 0.5 = Sobreviveu



Regressão Logística

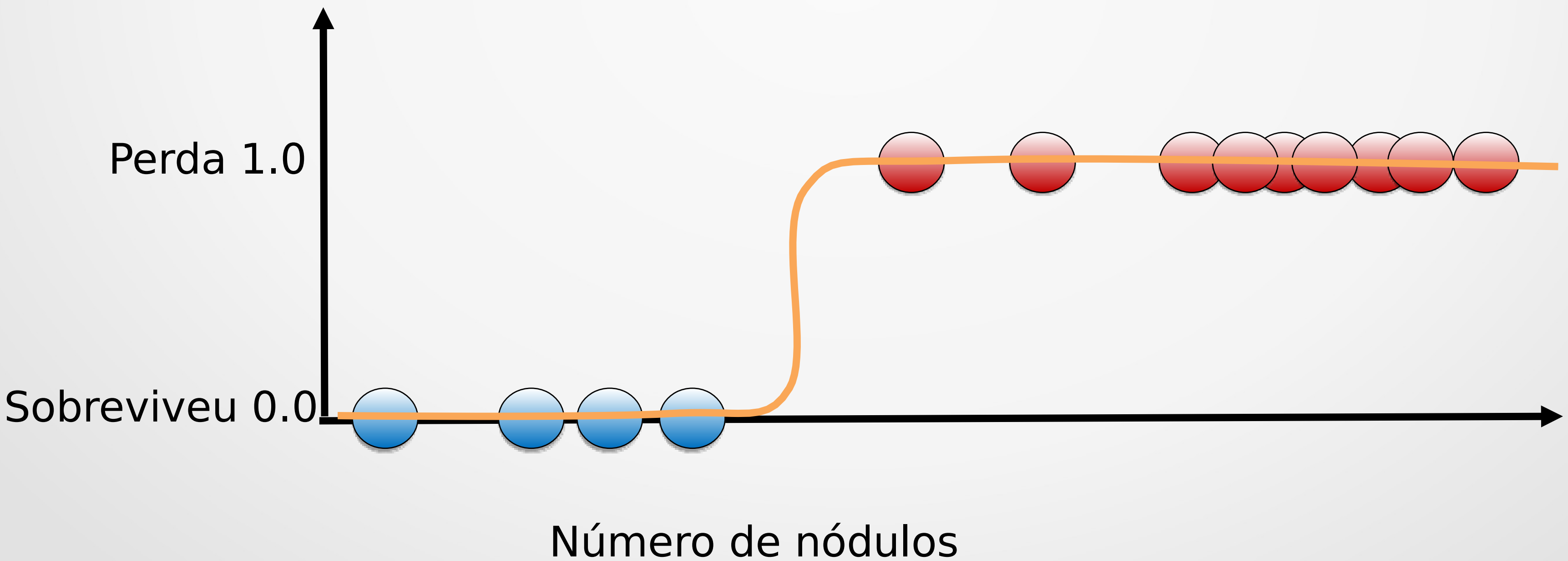
- Regressões não lineares são possíveis por meio do uso de outras funções, como a função logística



Regressão Logística

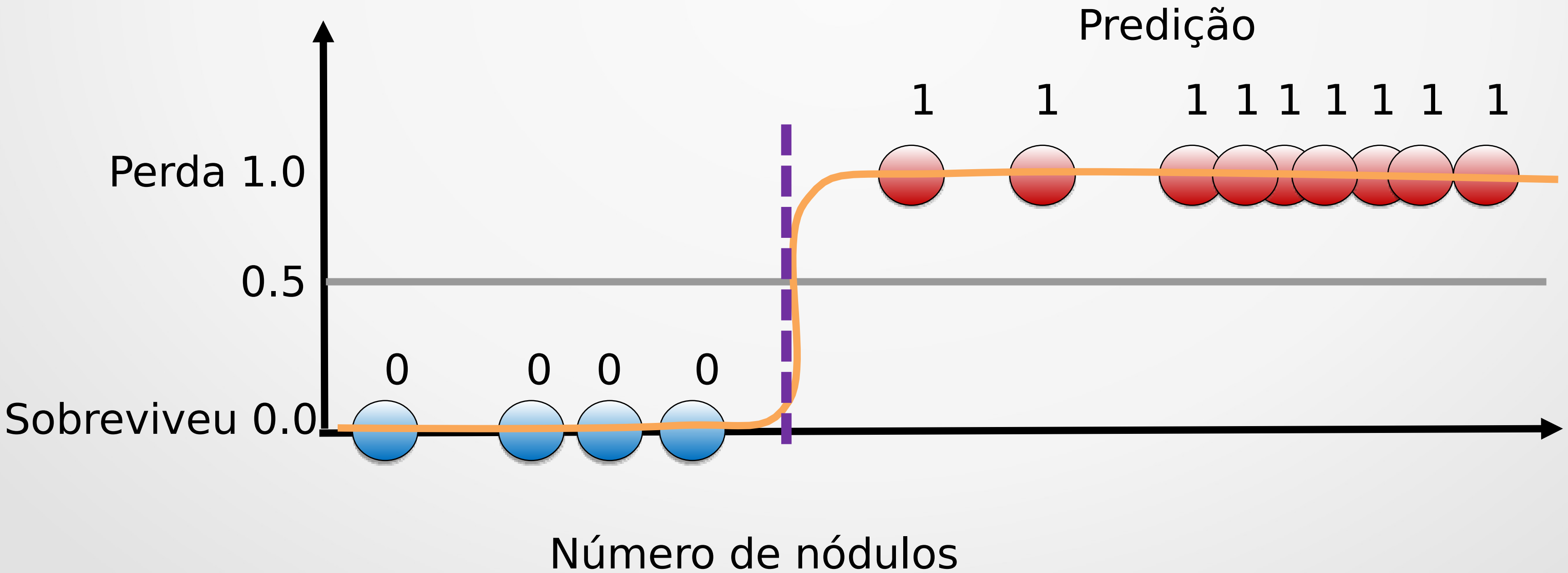
- Aplicando a função logística no exemplo anterior

$$y_\beta(x) = \frac{1}{1+e^{-(\beta_0+\beta_1 x + \varepsilon\ )}}$$



Regressão Logística

- Usando o seguinte limiar de decisão
  - Resultado do modelo > 0.5 = Perda
  - Resultado do modelo < 0.5 = Sobreviveu



Relação entre regressão linear e logística

- Existe uma relação entre a função logística e linear
  - Função logística é conhecida como função *logit*
    - Ou seja, log-odds, o logaritmo do odds de p/(1-p)

$$P(x) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x + \varepsilon)}}$$

$$P(x) = \frac{e^{(\beta_0 + \beta_1 x)}}{1 + e^{(\beta_0 + \beta_1 x)}}$$

Relação entre regressão linear e logística

- Função Logística $$P(x) = \frac{e^{(\beta_0 + \beta_1 x)}}{1+e^{(\beta_0 + \beta_1 x)}}$$

- Odds $$\frac{P(x)}{1 - P(x)} = e^{(\beta_0 + \beta_1 x)}$$

- Log Odds $$log\left[\frac{P(x)}{1 - P(x)}\right] = \beta_0 + \beta_1 x$$

Relação entre regressão linear e logística

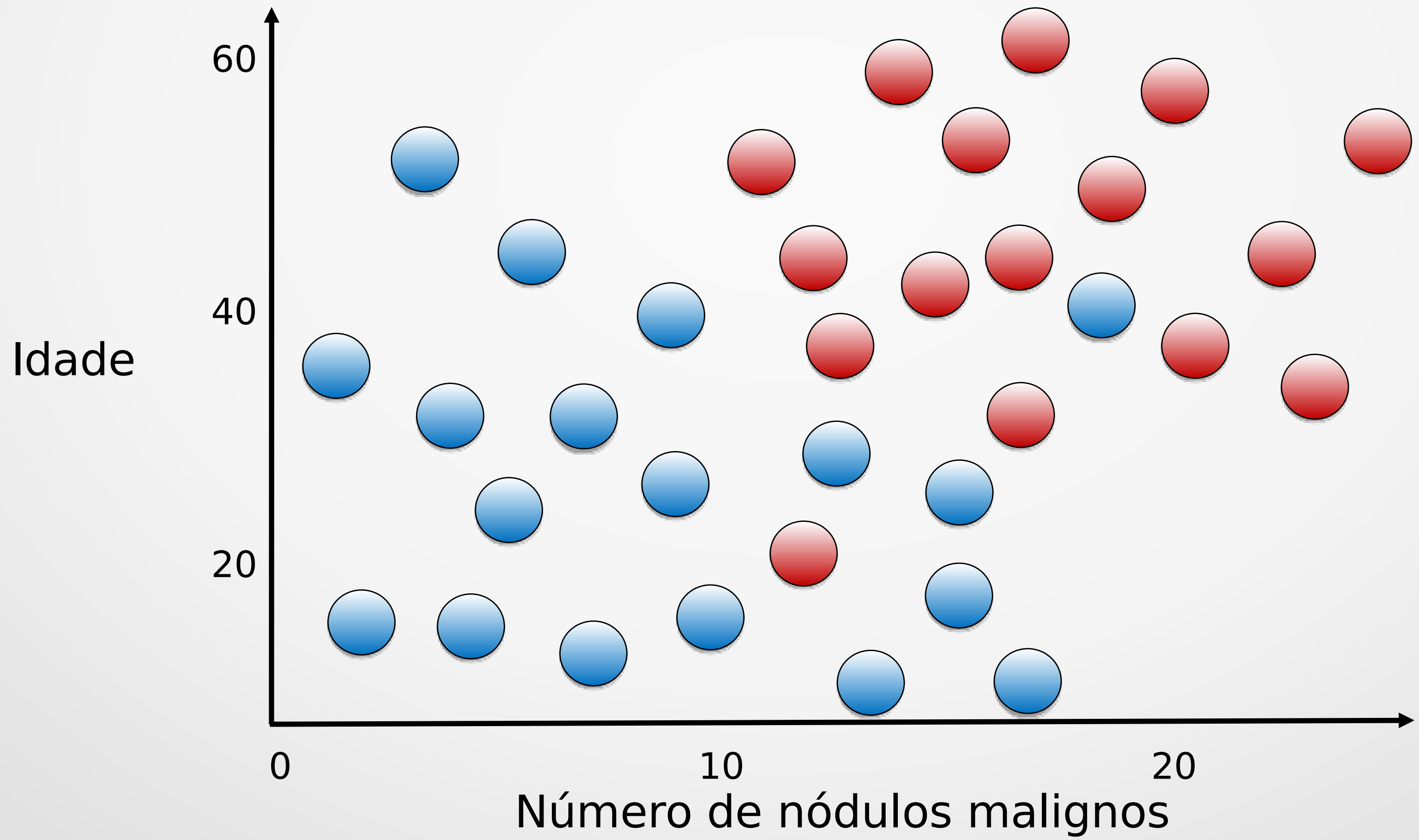
- Função Logística $$P(x) = \frac{e^{(\beta_0 + \beta_1 x)}}{1+e^{(\beta_0 + \beta_1 x)}}$$

- Odds $$\frac{P(x)}{1 - P(x)} = e^{(\beta_0 + \beta_1 x)}$$

- Log Odds $$log\left[\frac{P(x)}{1 - P(x)}\right] = \beta_0 + \beta_1 x$$

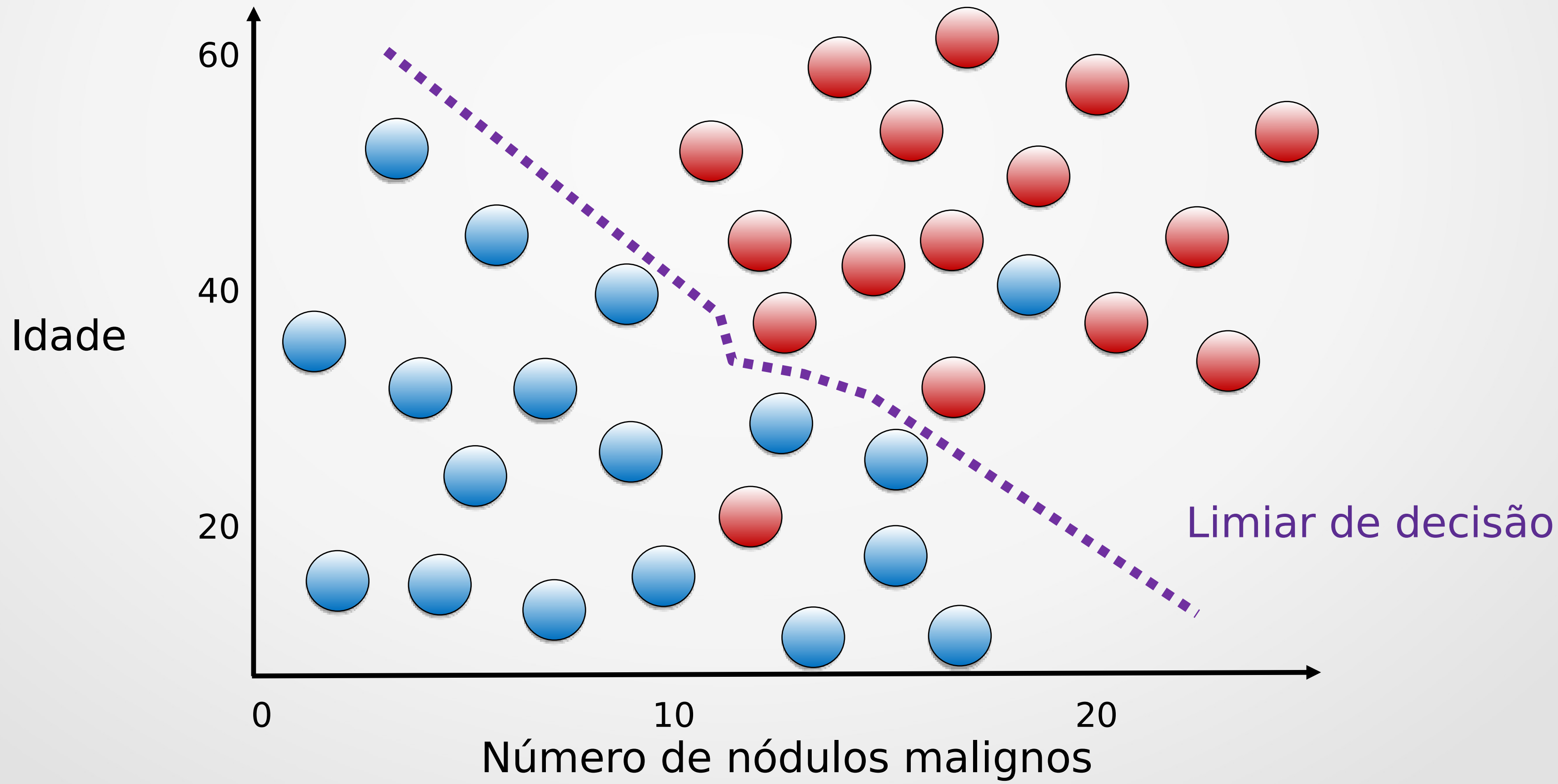
Classificação com Regressão

- Dois atributos (nódulos, idade)
- Dois rótulos (sobreviveu, perda)



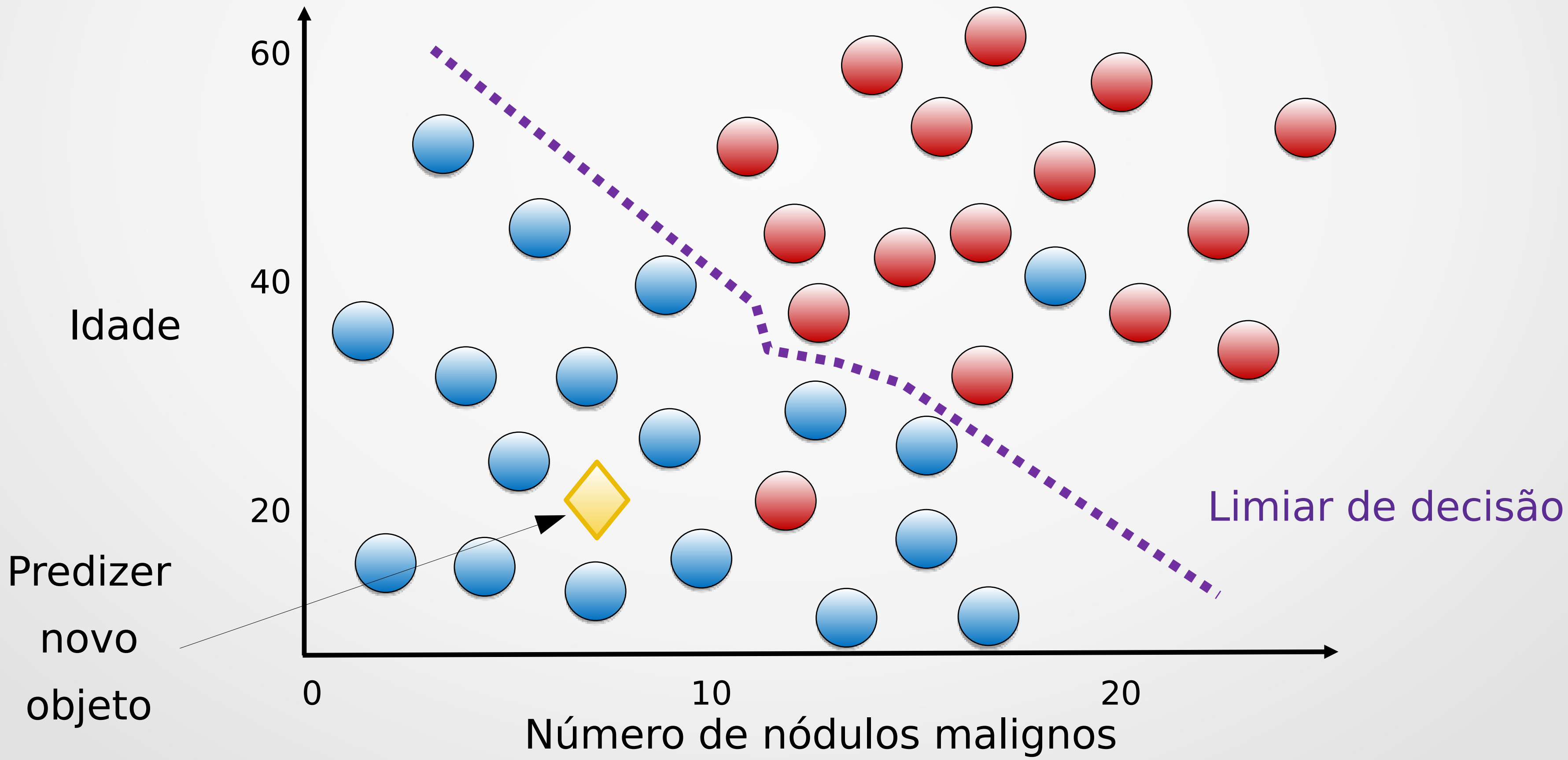
Classificação com Regressão

- Dois atributos (nódulos, idade)
- Dois rótulos (sobreviveu, perda)



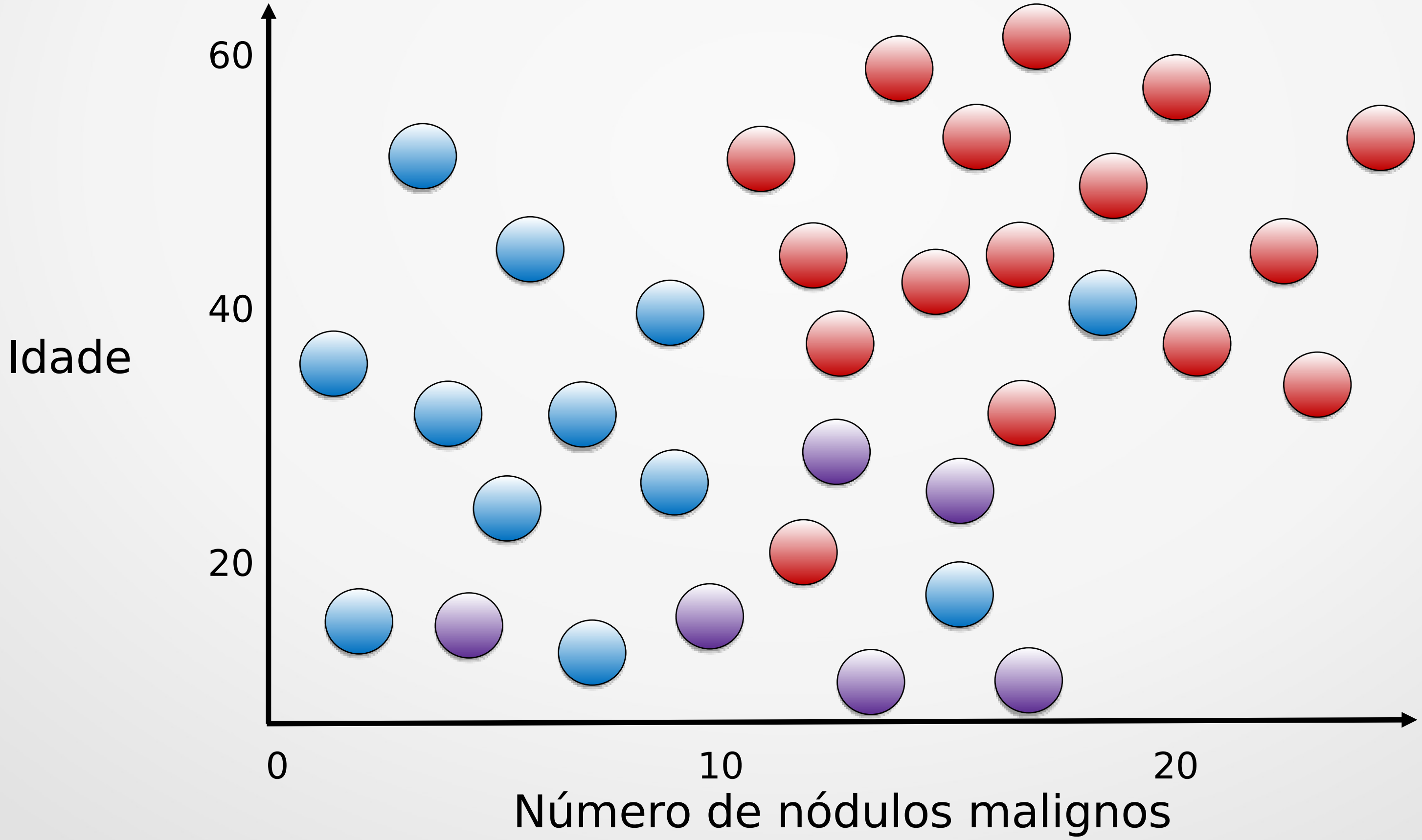
Classificação com Regressão

- Dois atributos (nódulos, idade)
- Dois rótulos (sobreviveu, perda)



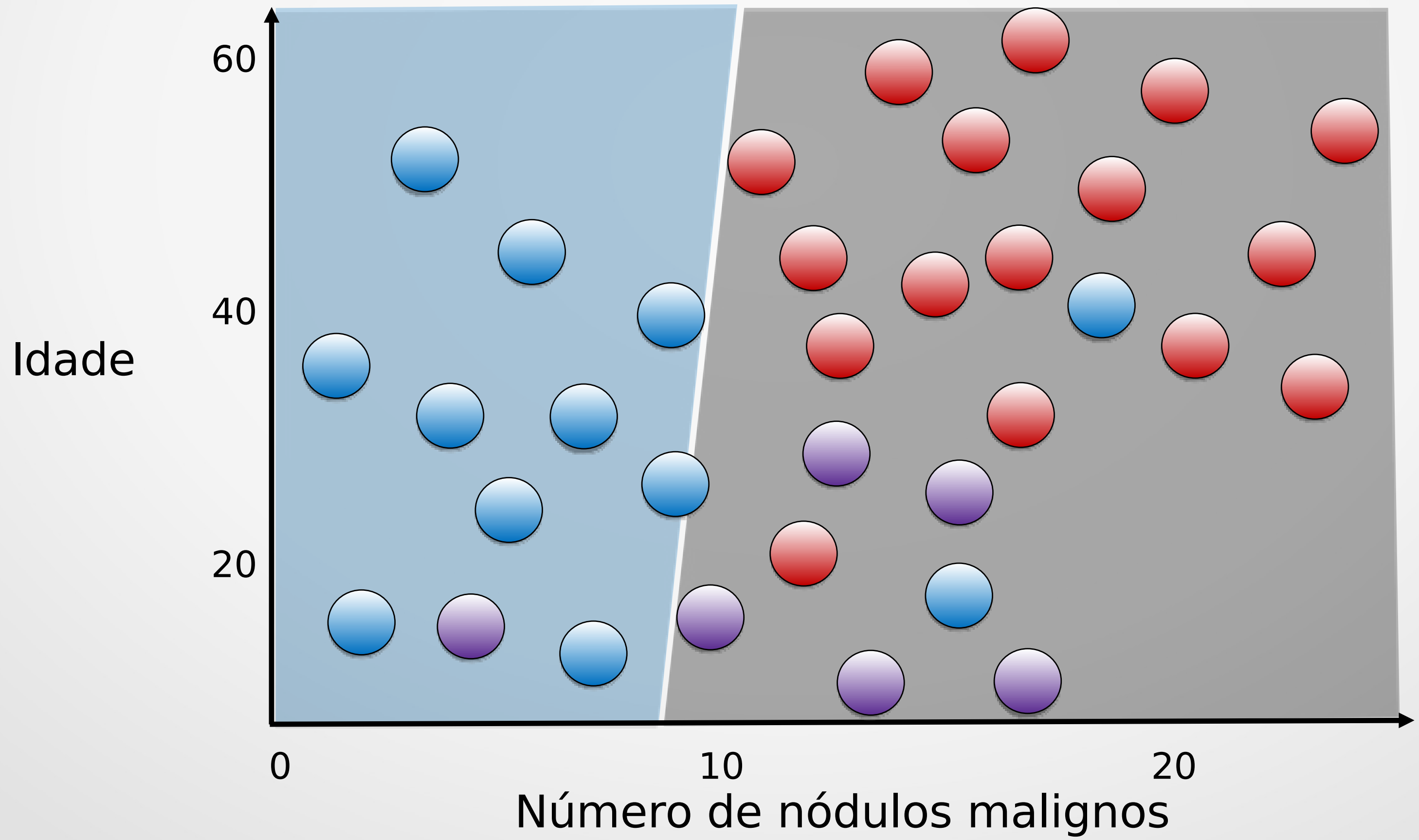
Classificação Multi-classe com Regressão

- Dois atributos (nódulos, idade)
- Três rótulos (sobreviveu, perda, complicações)



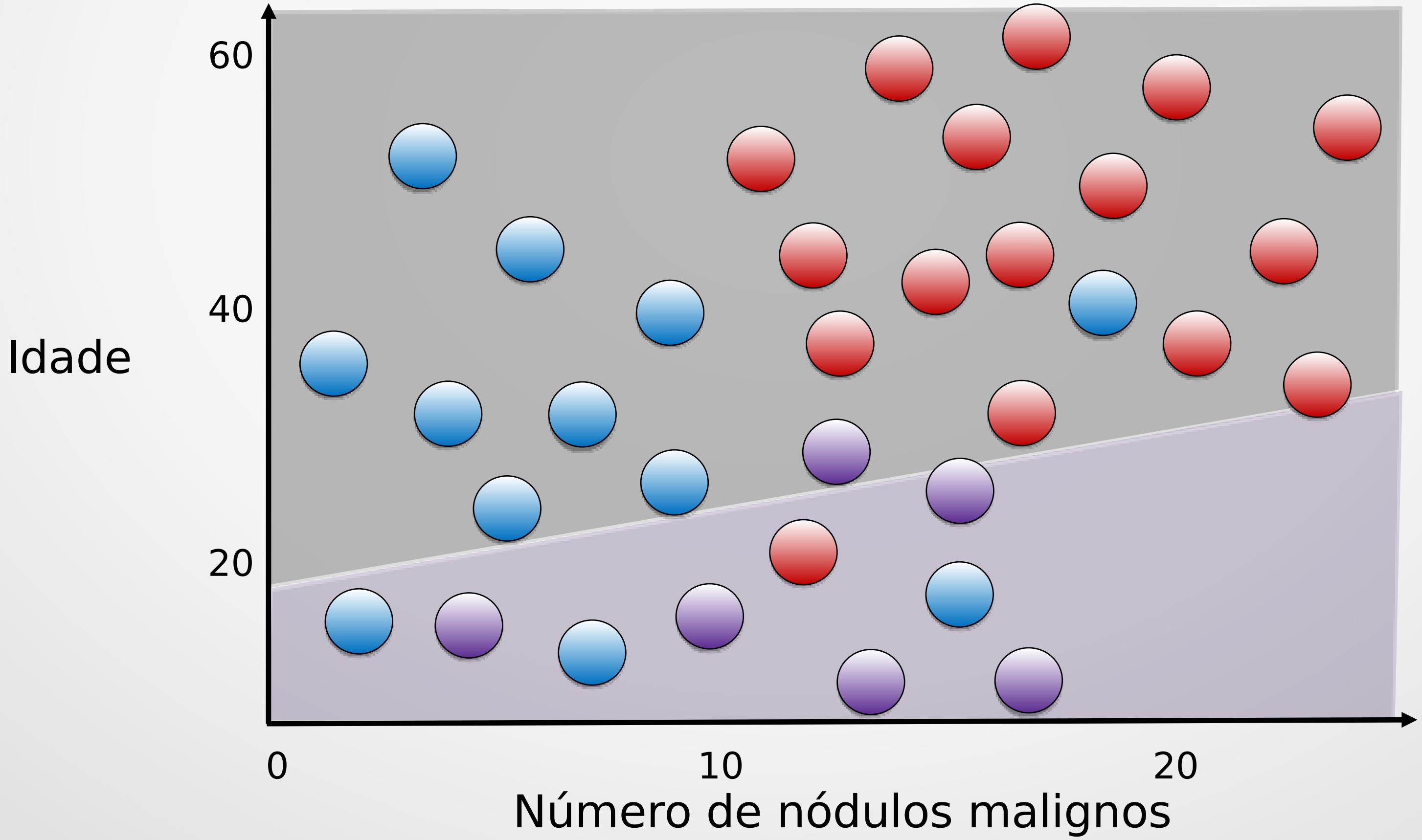
Um contra todos: sobreviveu

- Dois atributos (nódulos, idade)
- Três rótulos (sobreviveu, perda, complicações)



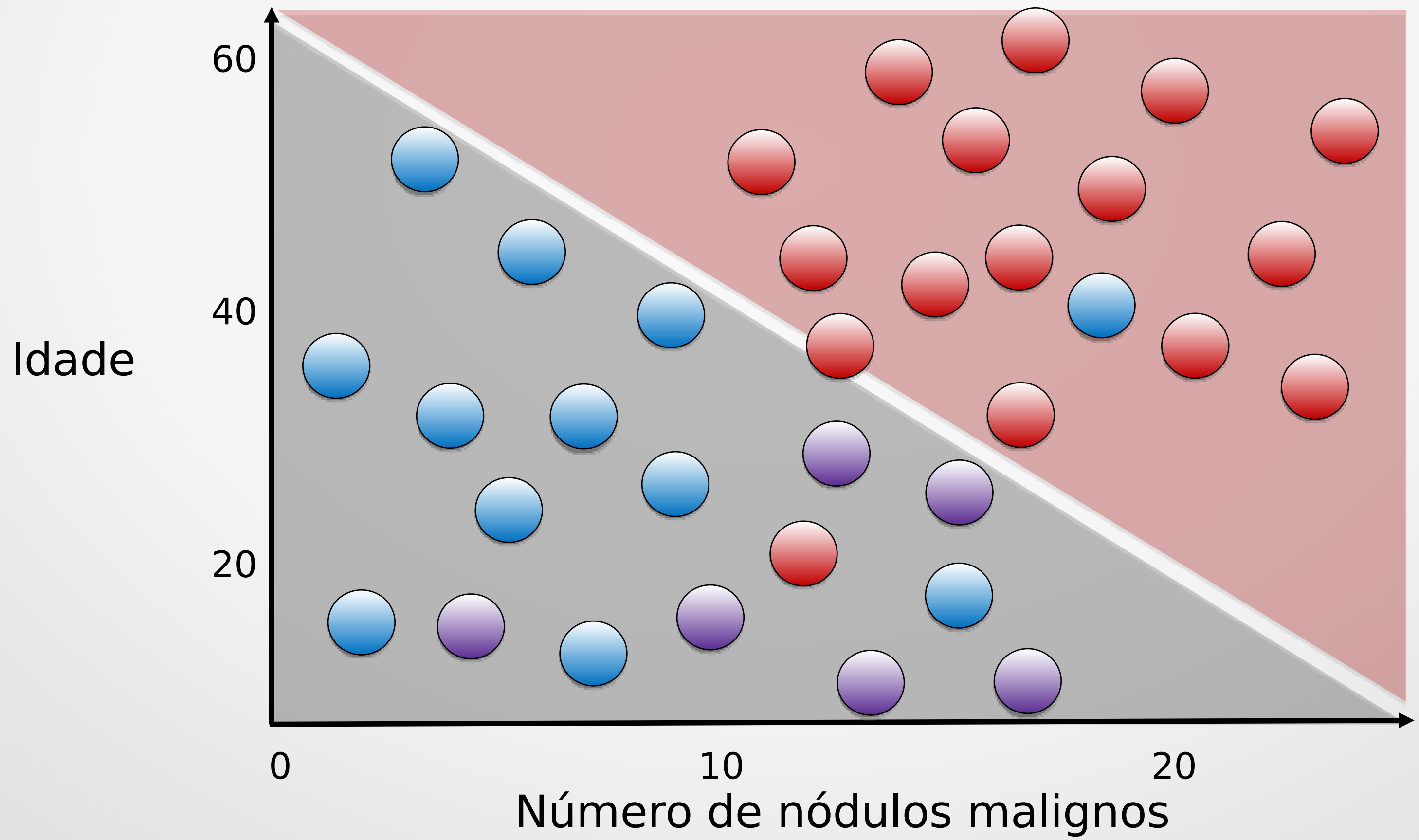
Um contra todos: complicações

- Dois atributos (nódulos, idade)
- Três rótulos (sobreviveu, perda, complicações)



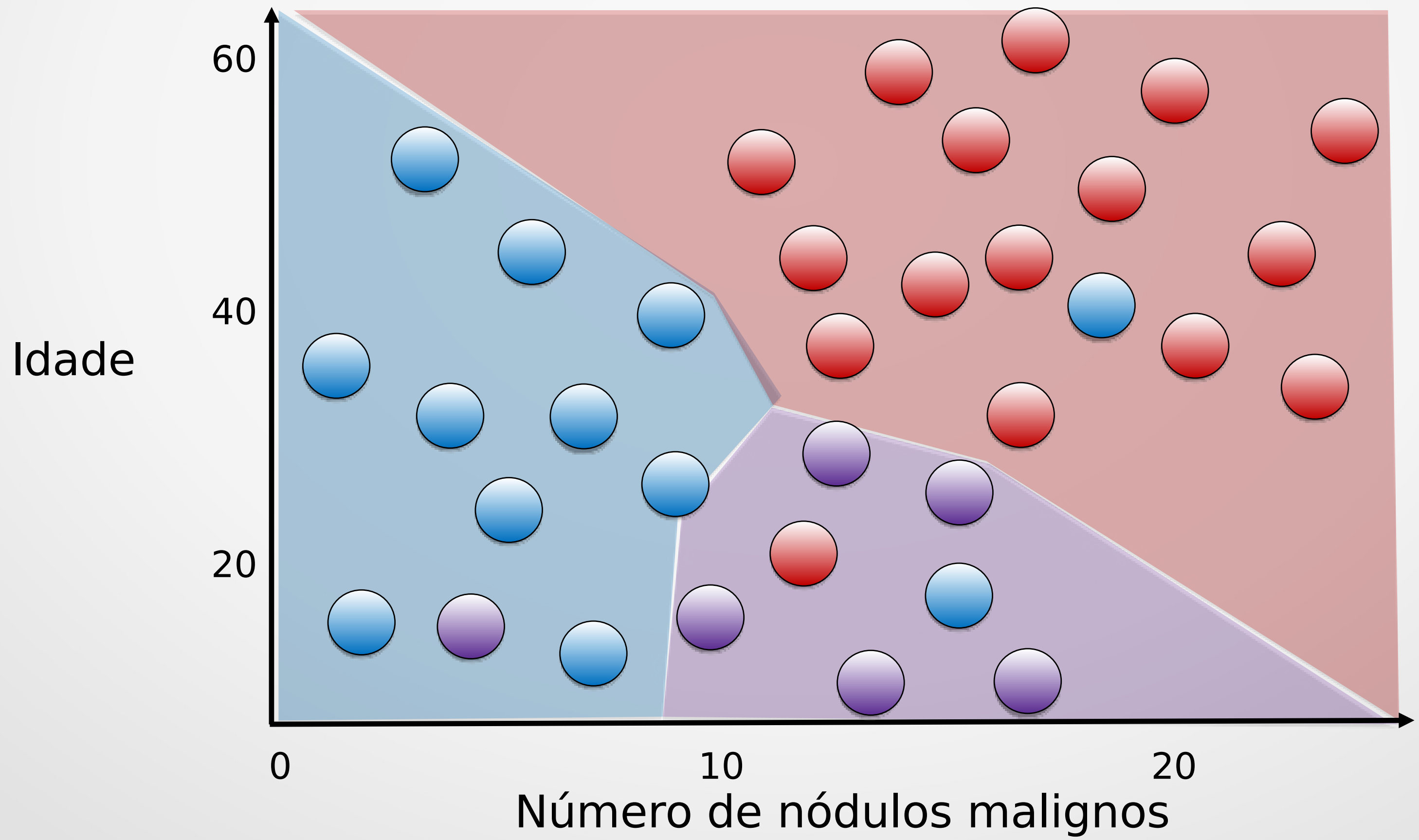
Um contra todos: perdas

- Dois atributos (nódulos, idade)
- Três rótulos (sobreviveu, perda, complicações)



Limiar de decisão multi-classe

- Dois atributos (nódulos, idade)
- Três rótulos (sobreviveu, perda, complicações)



Regressão Logística com *holdout* Iris

- Código

```
#Importando o método de Regressão Logística
from sklearn.linear_model import LogisticRegression
# Localização do arquivo
filepath = 'data/Iris_Data.csv'
# Importando os dados
data = pd.read_csv(filepath)
#Colocando os dados em ordem aleatória
randomdata = (data.sample(n=150, replace=False))
#Aplicando hold out
traindata = randomdata.iloc[:135,:]
testdata = randomdata.iloc[135:,:]
#Criando uma instância da classe
LR = LogisticRegression(penalty='l2',C=10.0)
#Faz o ajuste da classe aos dados de cimento
LR = LR.fit(traindata.iloc[:,0:4], traindata.iloc[:,4])
#Classe real
print(testdata.iloc[:,4])
#Classe predita
print(LR.predict(testdata.iloc[:,0:4]))
```

Regressão Logística com *holdout* Iris

- Saída = Classes

```
35              Iris-setosa
96          Iris-versicolor
114          Iris-virginica
132          Iris-virginica
63          Iris-versicolor
21              Iris-setosa
12              Iris-setosa
0               Iris-setosa
36              Iris-setosa
146          Iris-virginica
73          Iris-versicolor
50          Iris-versicolor
41              Iris-setosa
144          Iris-virginica
75          Iris-versicolor
```

- Saída Predita

```
['Iris-setosa'
'Iris-versicolor'
'Iris-virginica'
'Iris-virginica'
 'Iris-versicolor'
'Iris-setosa'
'Iris-setosa'
'Iris-setosa'
'Iris-setosa'
 'Iris-virginica'
'Iris-versicolor'
'Iris-versicolor'
'Iris-setosa'
 'Iris-virginica'
'Iris-versicolor']
```