

MBA⁺

**ARTIFICIAL INTELLIGENCE
& MACHINE LEARNING**

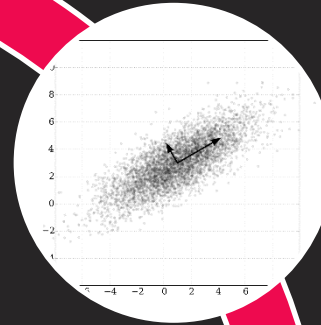
MBA⁺

MODELOS DE MACHINE LEARNING COM R

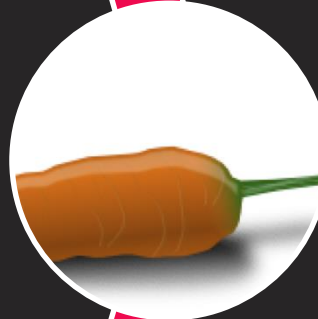
Prof. Elthon Manhas de Freitas
elthon@usp.br

2018

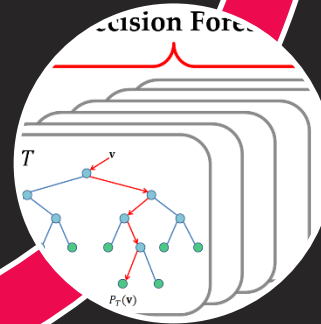
Agenda de hoje



PCA



Caret
Package



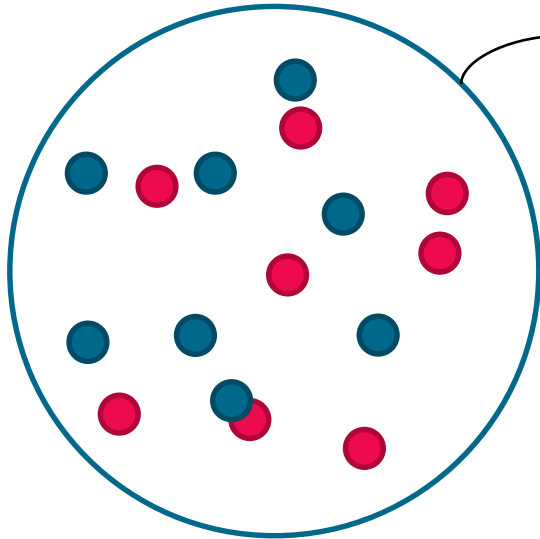
Árvores

Predição

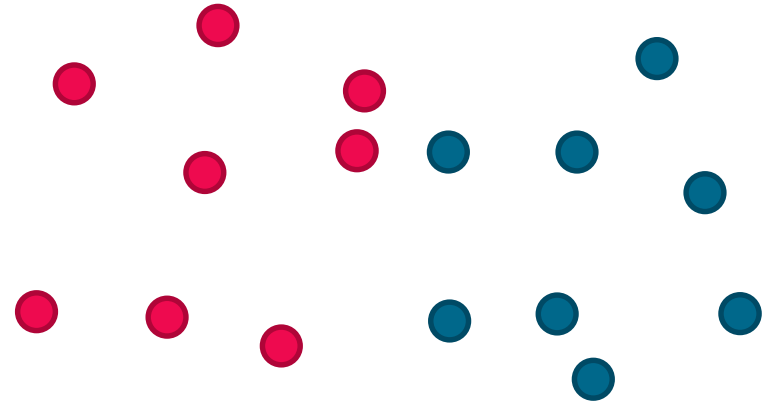


Como funciona a predição

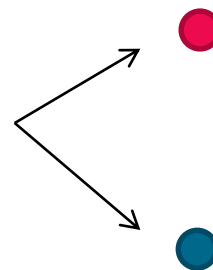
Amostra da população



Conjuntos de treinamento



$f(o) =$



Função de predição

Matriz de confusão

Será visto com detalhes em Machine Learning / Deep Learning

		Resposta correta	
		Positivo	Negativo
Predito	Positivo	Verdadeiro Positivo	Falso Positivo
	Negativo	Falso Negativo	Verdadeiro Negativo



Acurácia

$$- \frac{\text{Total de acertos}}{\text{População}}$$



Sensibilidade (*Precisão*)

$$- \frac{\text{Acertos Positivos}}{\text{Total de Positivos}}$$



Especificidade

$$- \frac{\text{Acertos Negativos}}{\text{Total de Negativos}}$$



Eficiência

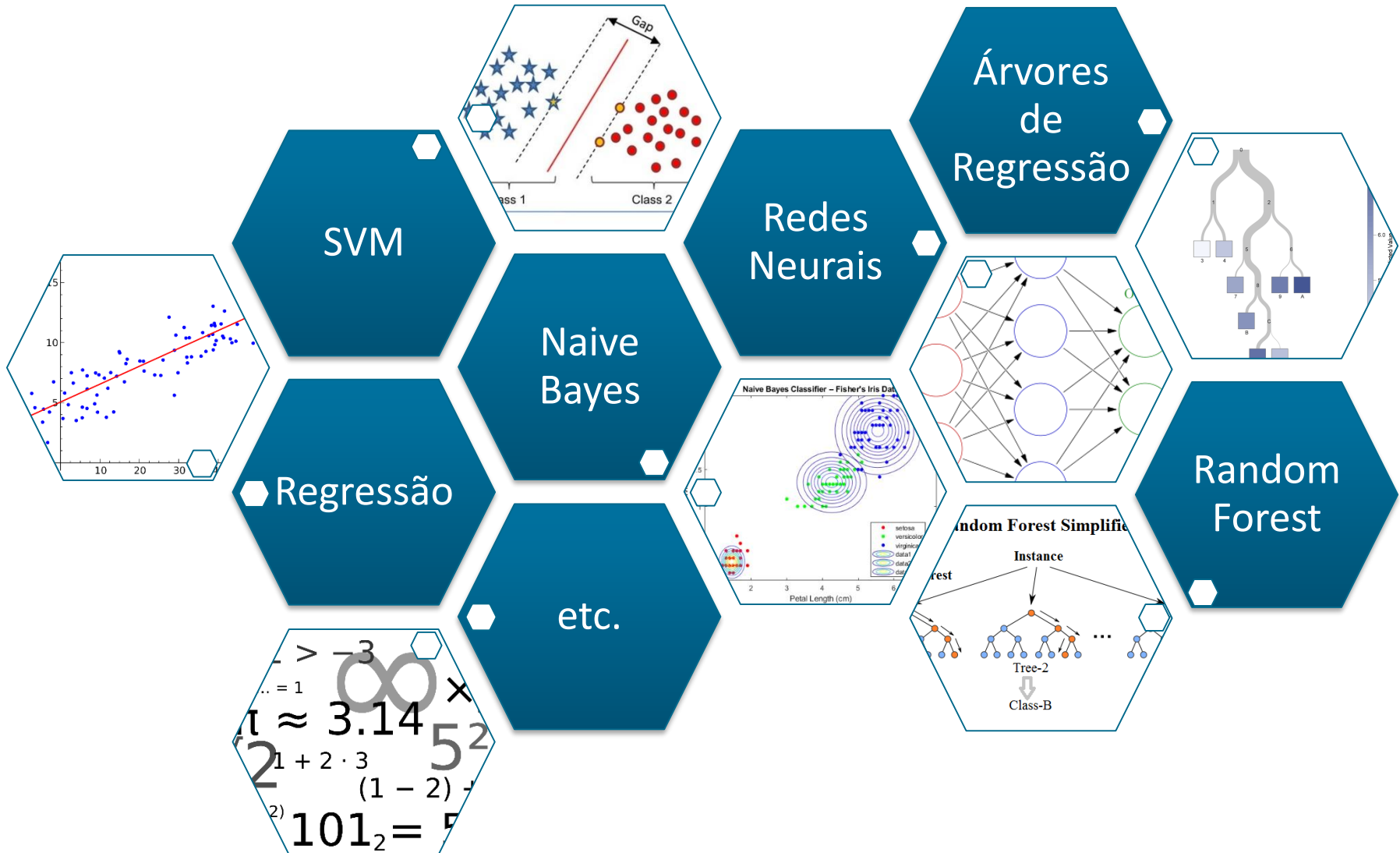
$$- \frac{\text{Sensibilidade} + \text{Especificidade}}{2}$$

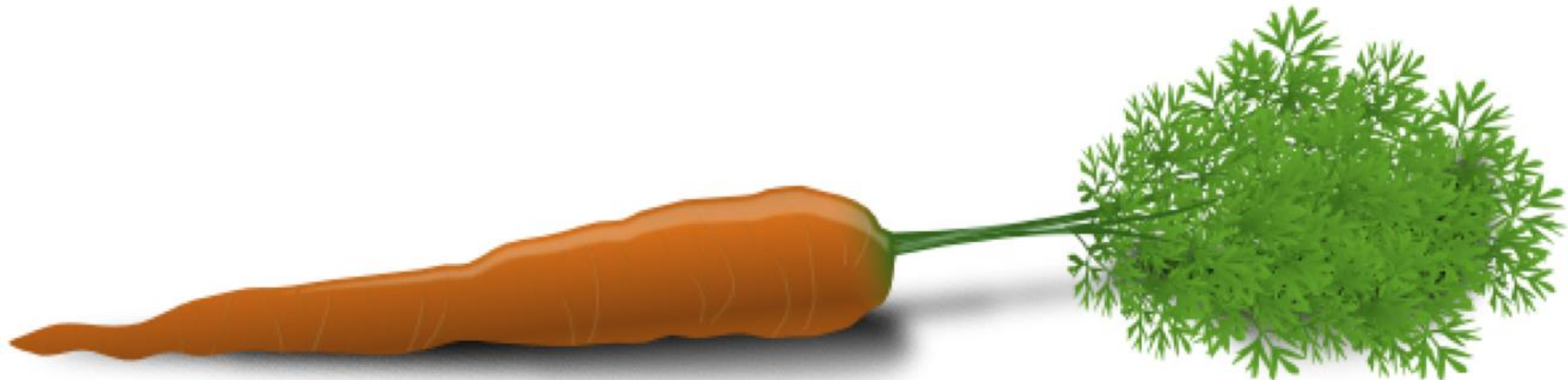


Recall

$$- \frac{\text{Acertos Positivos}}{\text{Acertos Positivos} + \text{Falso Negativo}}$$

Técnicas e algoritmos de predição





The caret package (short for **C**lassification **A**nd **RE**gression **T**raining) is a set of functions that attempt to streamline the process for creating predictive models. The package contains tools for:

237 modelos (jun/2018)


- data splitting
- pre-processing
- feature selection
- model tuning using resampling
- variable importance estimation

- <http://topepo.github.io/caret/index.html>

- <http://cran.r-project.org/web/packages/caret/index.html>

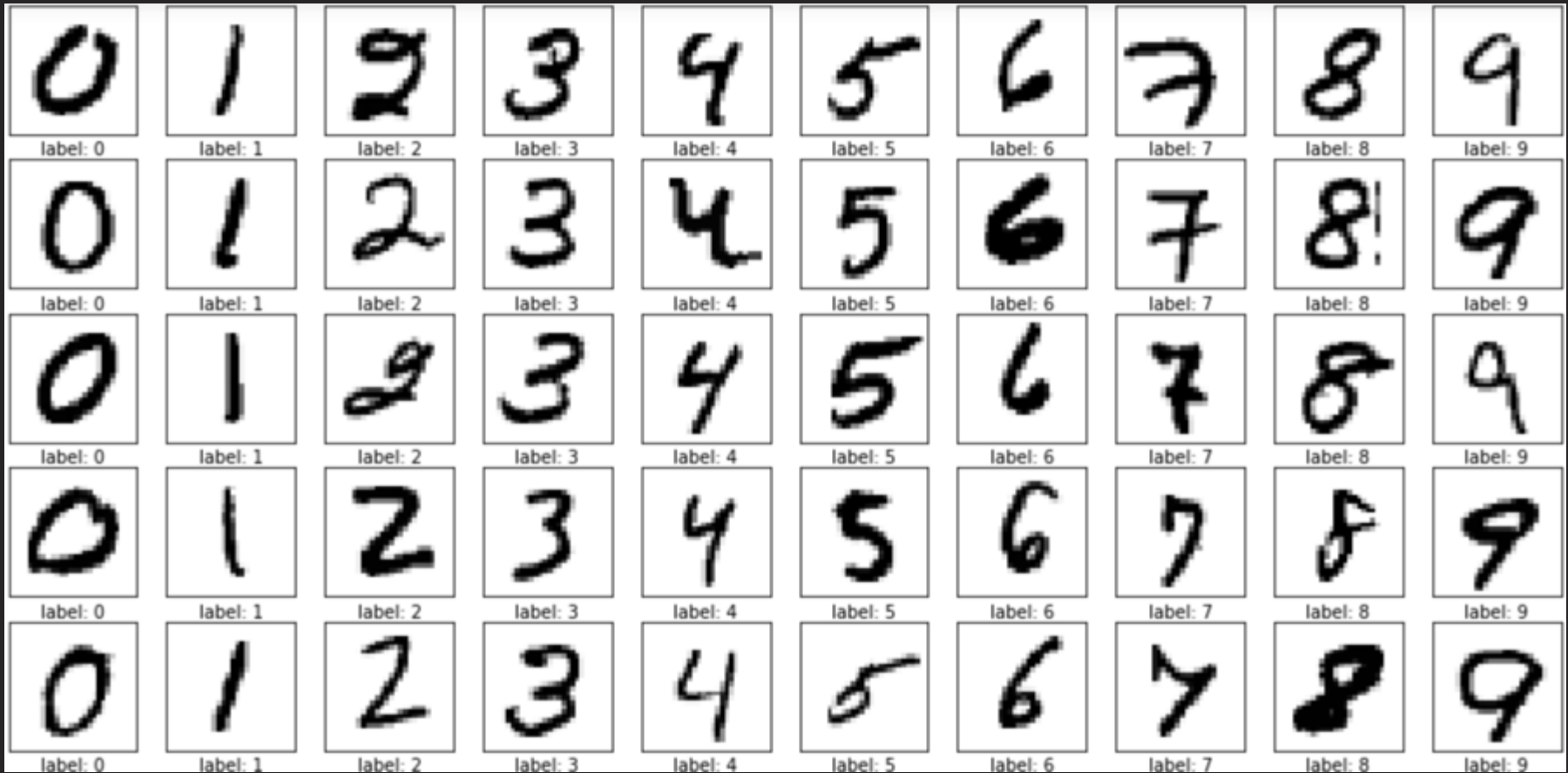
Regressão linear com PCA

- Vamos continuar com o dataset spam visto há pouco
- Etapas para a modelagem de uma predição:
 - Definir entradas
 - (Feature Engineering)
 - Definir saída(s)
 - Definir processamento
 - Avaliação
- Entrada:
 - PCA
 - Saída: Spam ou Not-Spam
 - Processamento: Regressão linear
 - Avaliação: Matriz de confusão

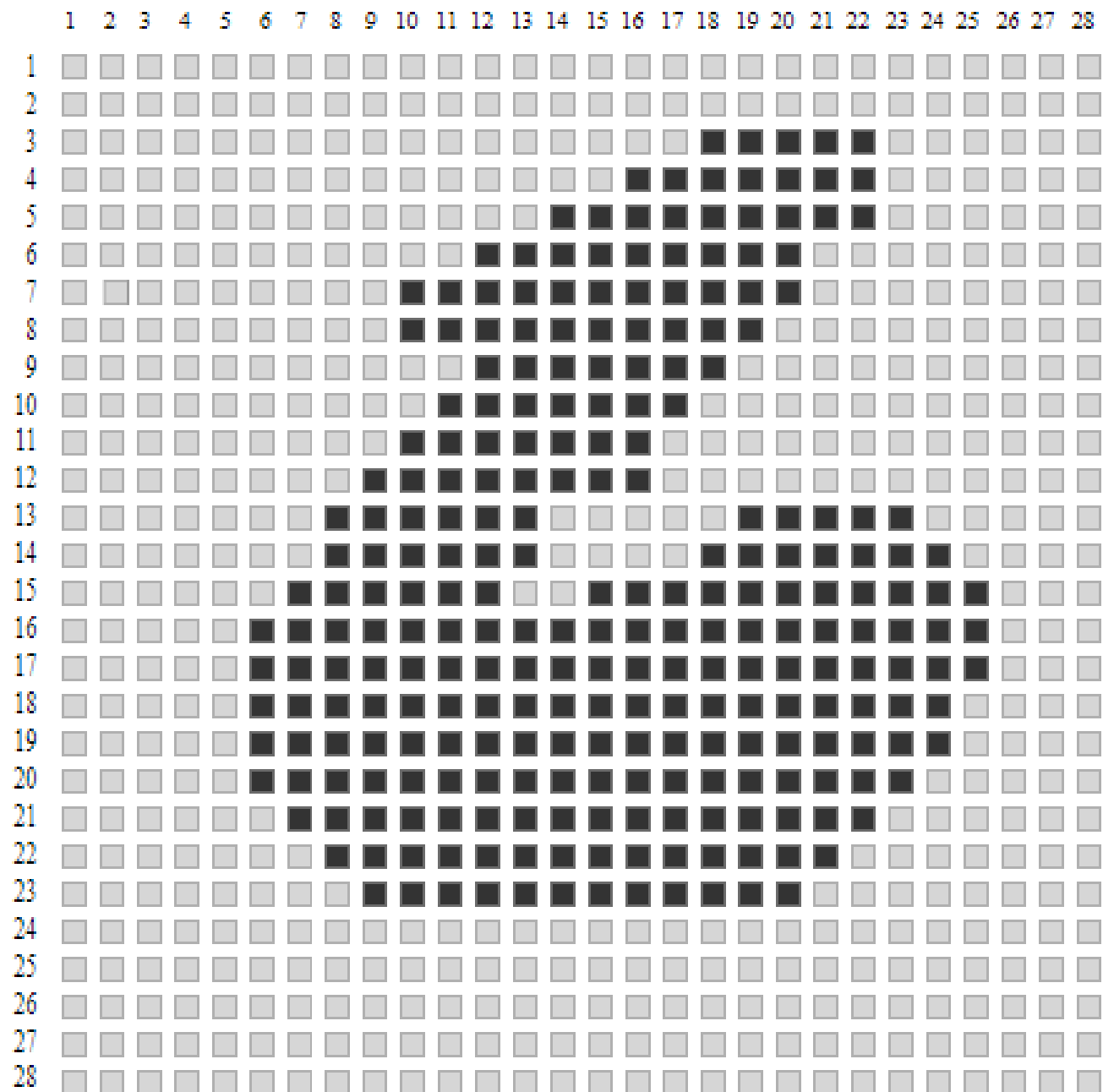


Demonstração
e
discussão
R-Markdown

THE MNIST DATABASE (Exploração)



<http://yann.lecun.com/exdb/mnist/>



$$28 \times 28 = 784$$

THE MNIST DATABASE (Exploração)

- Carregar o dataset: `mnist.RData`
- Explorar o dataset
 - Função de consulta:

```
display_number <- function( x = NA, dataset = train){  
  if ( is.na(x)){x <- sample(seq_len(dataset$n), size = 1)}  
  
  image( matrix( dataset$x[x, ], ncol = 28),  
         axes = F, col= gray(255:1/255))  
  
  box()  
  title( paste0(x, ': ', dataset$y[x]) )  
}  
#Exemplos:  
display_number()  
display_number(x=3189)  
display_number(dataset = test)  
display_number(dataset = test, x=3189)
```

THE MNIST DATABASE (Predição)

- O professor já preparou o dataset 😊
- Faremos:

Treinamento do modelo



Predição do modelo



Bônus!! Predição simples

- Notebook com exemplo de predição de salário em função da idade
 - Utilizando regressão linear simples: l_m
 - Utiliza composição de características
 - (feature engineering)



PCA – Principal Component Analysis

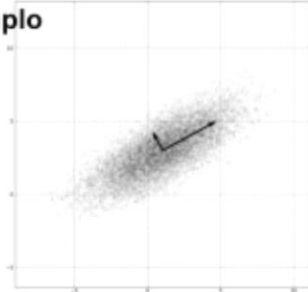
PCA – Principal Component Analysis

FIAP

- A técnica mais popular de seleção de características
- Inventado em 1901 por Karl Pearson



PCA - Exemplo



PCA de uma **distribuição Gaussiana multivariada** centrada em (1,3) com um desvio padrão de 3 aproximadamente na direção (0.878, 0.478) e desvio padrão 1 na direção ortogonal. Os vetores na figura são os **autovetores** da **matriz de covariância** multiplicados pela raiz quadrada do **autovalor** correspondente, e transladados de forma a iniciarem na média.

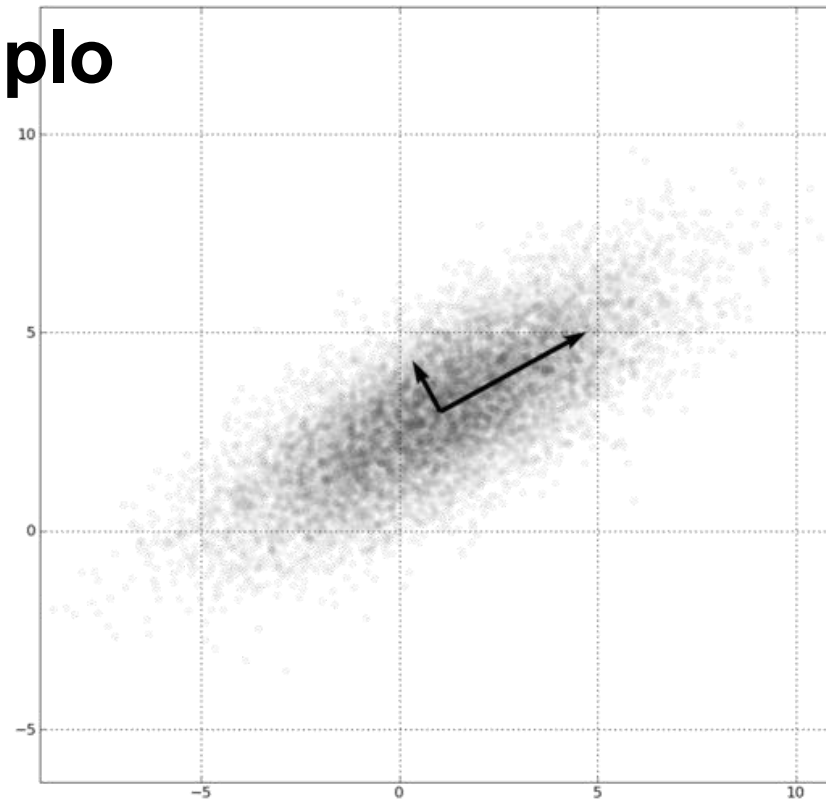
Fonte: Wikipedia

https://pt.wikipedia.org/wiki/An%C3%A1lise_de_componentes_principais

- Calculado por Autovetores e Autovalores de uma matriz de covariância

Muito usado em Computer Vision

PCA - Exemplo



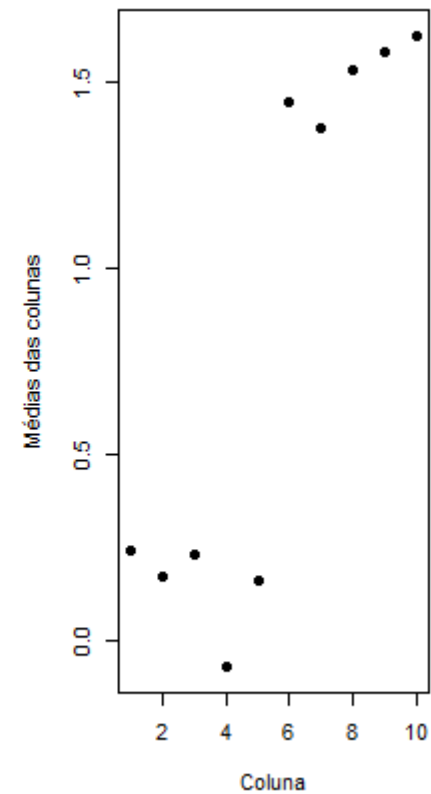
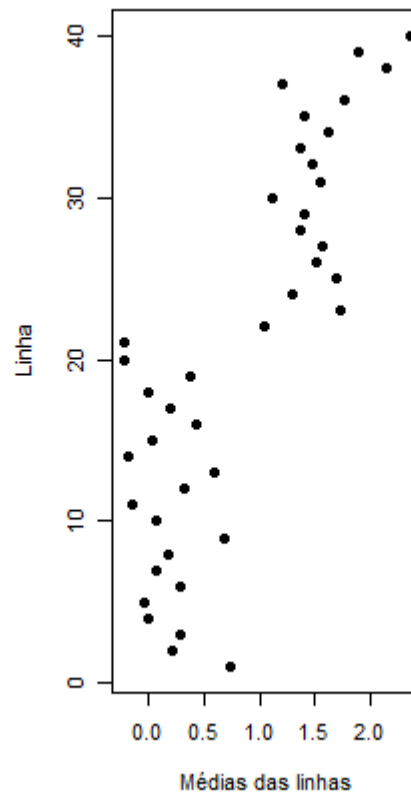
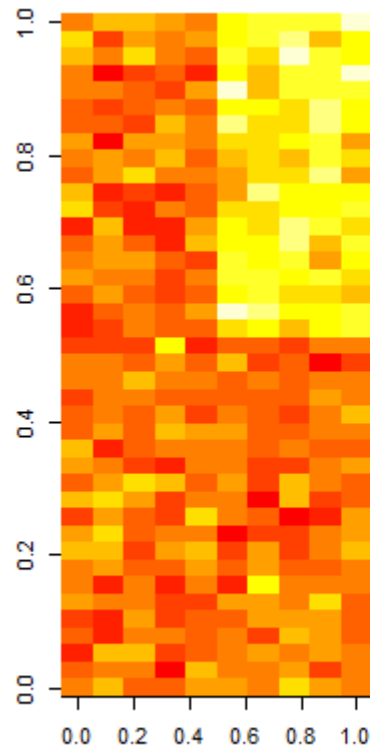
PCA de uma **distribuição Gaussiana multivariada** centrada em (1,3) com um desvio padrão de 3 aproximadamente na direção (0.878, 0.478) e desvio padrão 1 na direção ortogonal. Os vetores na figura são os **autovetores** da **matriz de covariância** multiplicados pela raiz quadrada do **autovalor** correspondente, e transladados de forma a iniciarem na média.

Fonte: Wikipedia

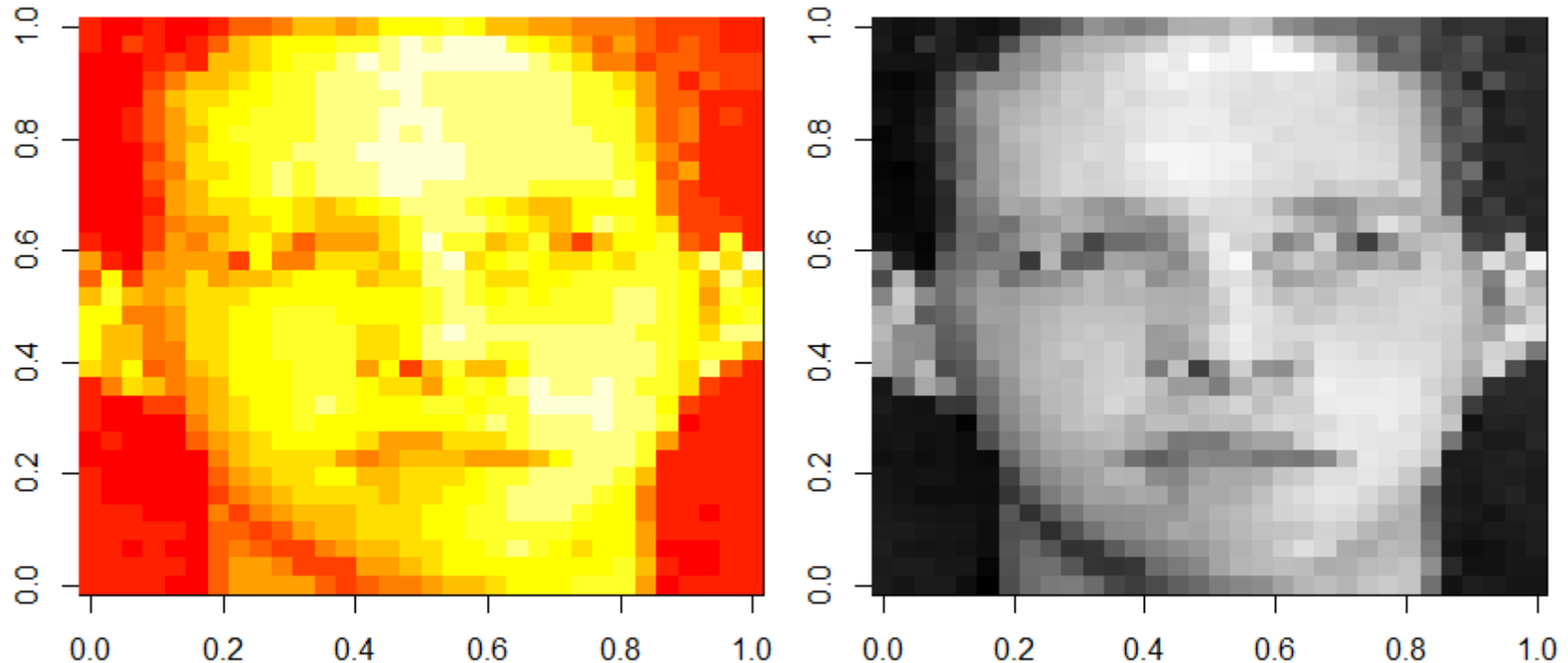
https://pt.wikipedia.org/wiki/An%C3%A1lise_de_componentes_principais

PCA no R

Consultar Rmarkdown de PCA



Como o autovetor consegue representar diversos valores



```
rosto <- read.csv("data/elthon.csv", header = FALSE) %>% as.matrix()
```

Como a imagem é 32x32 teremos:

- 32 linhas representando as observações

- 32 colunas representando as características

Criando o modelo PCA

Exibir a imagem

```
rosto <- read.csv("data/elthon.csv", header = FALSE) %>% as.matrix()

cinzas = grey(seq(0, 1, length = 256))
image( rosto )
image( rosto, col = cinzas )
```

Extrair as principais características

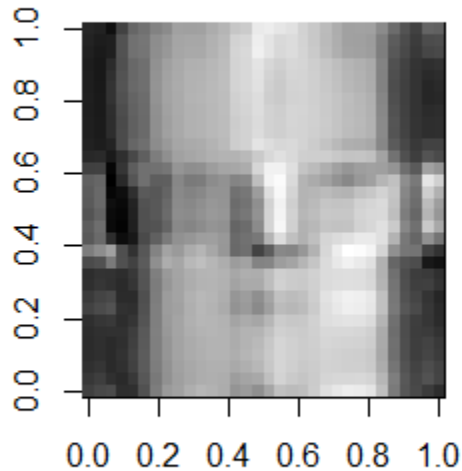
Quantos auto-vetores teremos?

```
pca.rosto <- prcomp( rosto, scale. = TRUE )
```

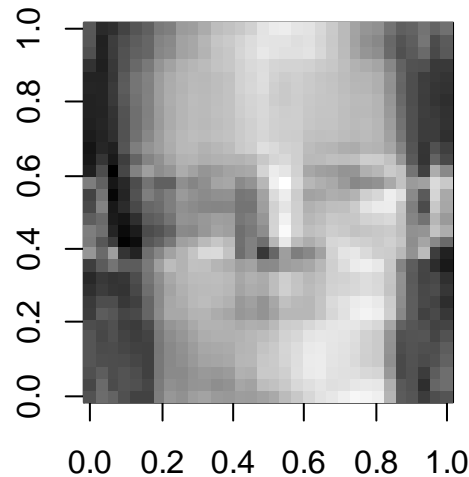
Cada autovetor fica armazenado em uma coluna do atributo rotation do modelo
(ex.: `pca.rosto$rotation`)

Restaurando os dados originais

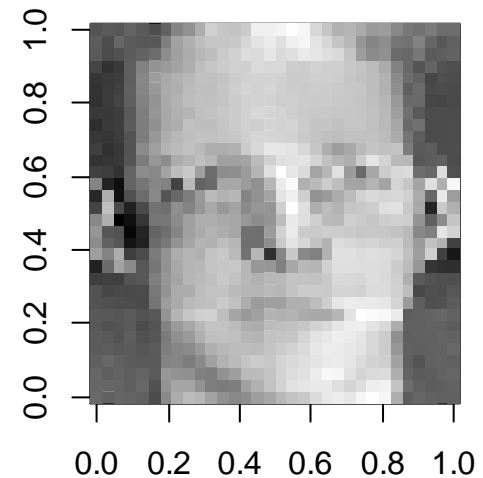
```
nd = 3  
image(pca.rosto$x[,1:nd] %*% t(pca.rosto$rotation[,1:nd]),  
      col = cinzas)
```



3 dimensões

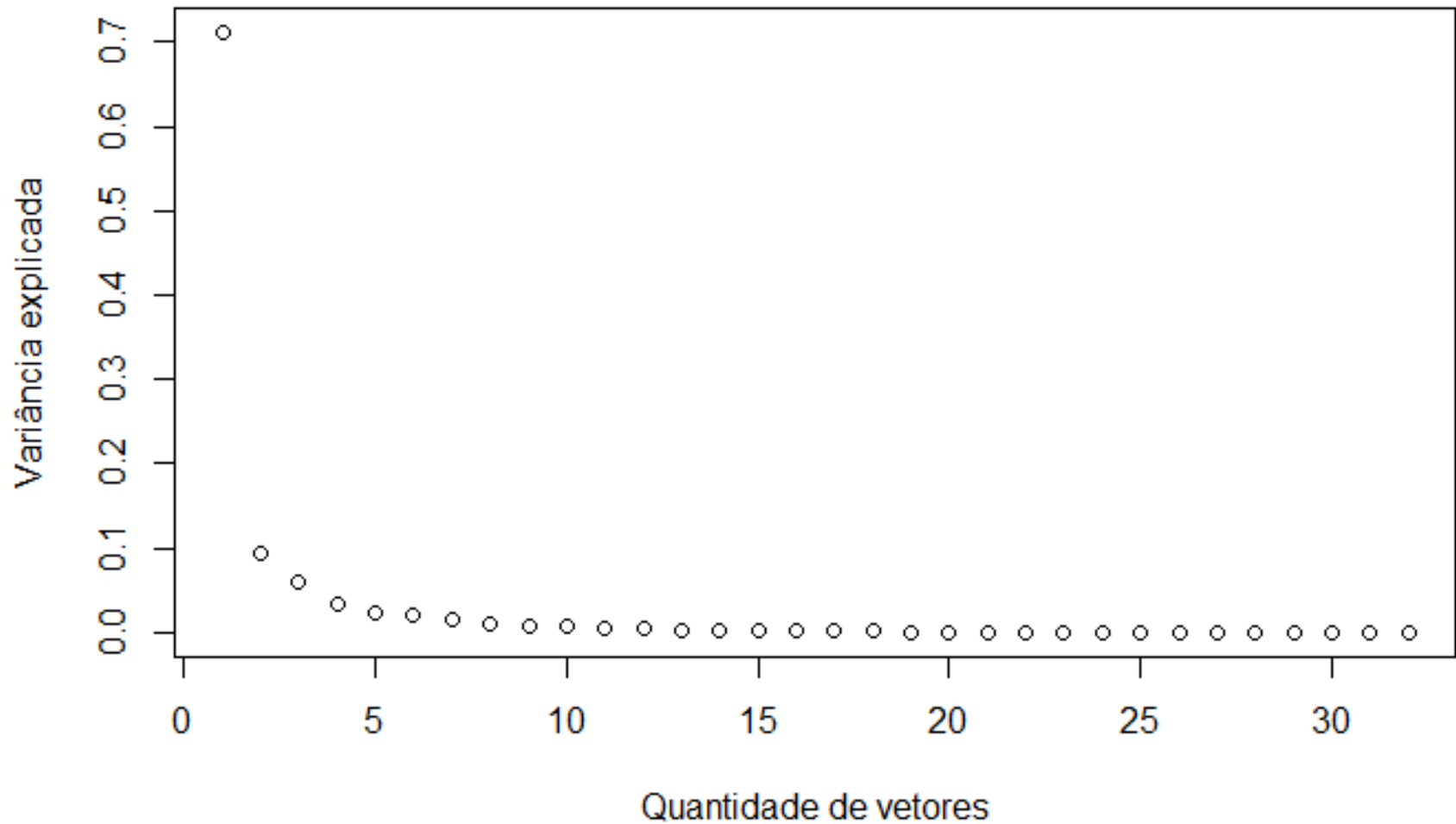


5 dimensões



15 dimensões

Análise da quantidade de vetores



Um alternativa ao PCA → SVD

singular value decomposition

PCA

- As principais características são extraídas e mantidas em autovetores complementares

SVD

- mantém todos os vetores e valores originais para uma extração perfeita
- Necessita de um vetor complementar para armazenar as perdas (erros)

Código em R-Mark,
apenas demonstração

- Carregar o dataset spam

```
library(caret)
library(kernlab)
data(spam)
```

- A coluna type (coluna 58) possui o tipo (é ou não spam)
- Identificar as colunas com correlação acima de 80%
- Criar um mini-dataset com duas colunas que serão comprimidas
- Criar um modelo de compressão com a função `prcomp`
- Plotar cada coluna do modelo criado



Árvores e Florestas

IRIS Dataset

```
plot_ly(data = iris, x = ~Sepal.Length,  
        y = ~Sepal.Width, color = ~Species)
```



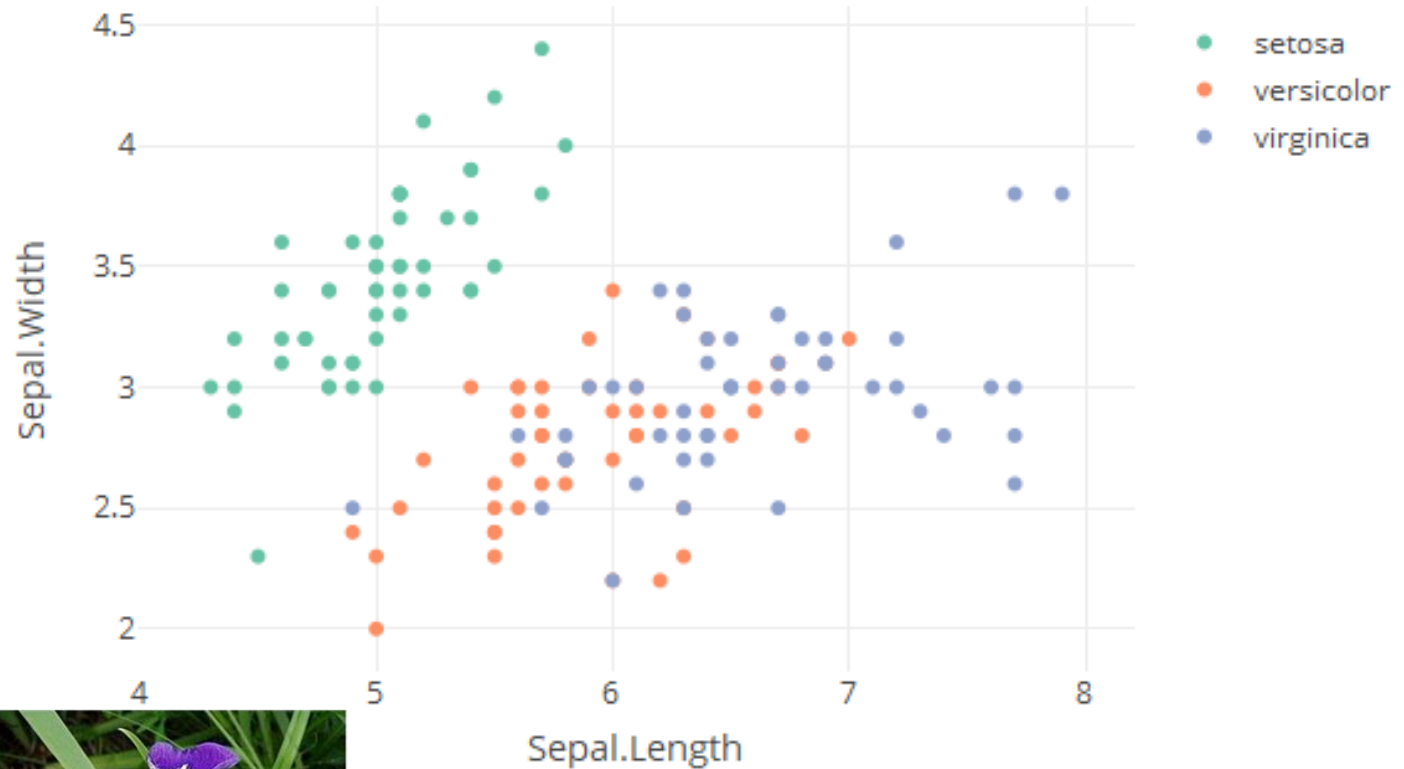
Iris Setosa



Iris Virginica



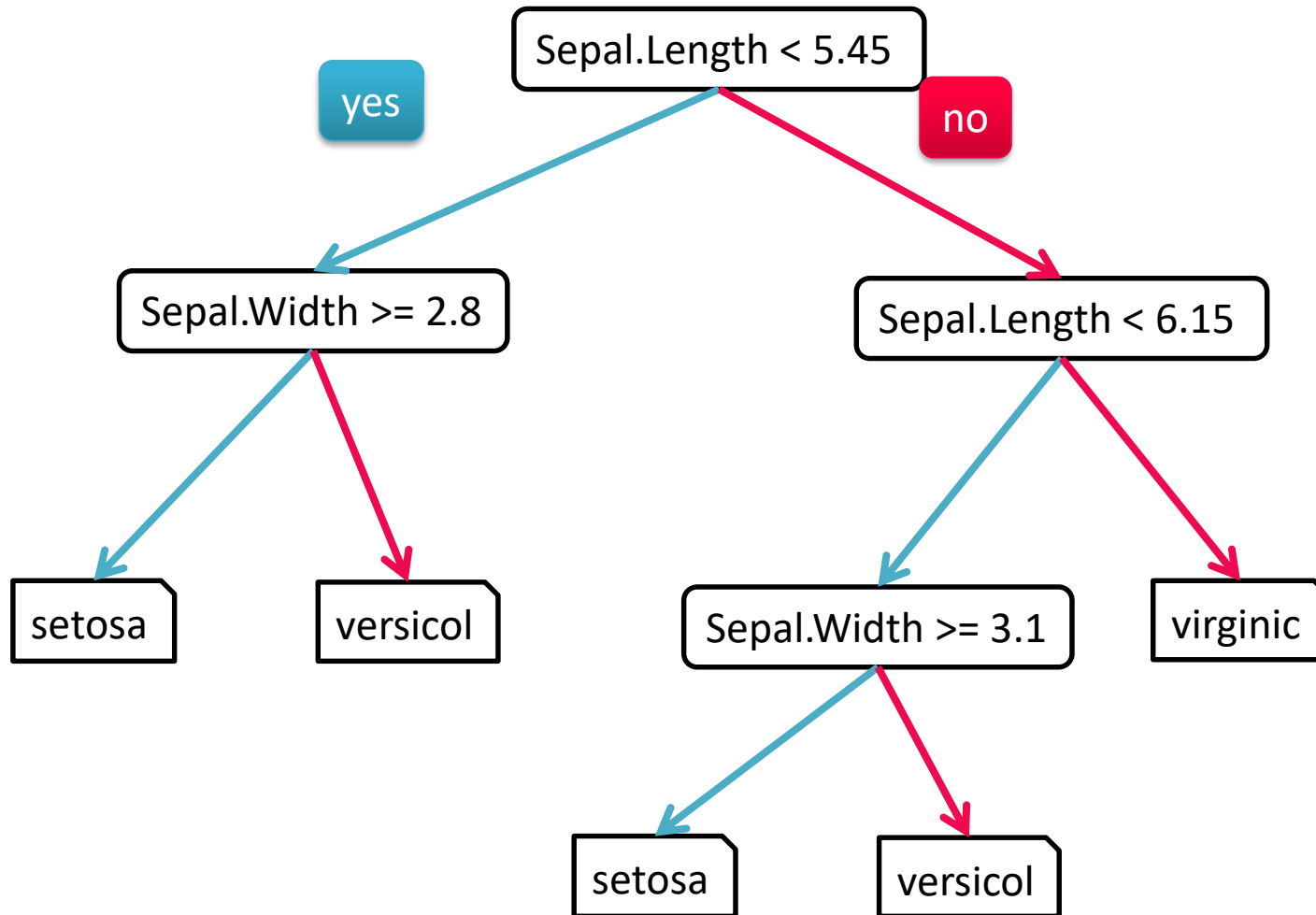
Iris Versicolor



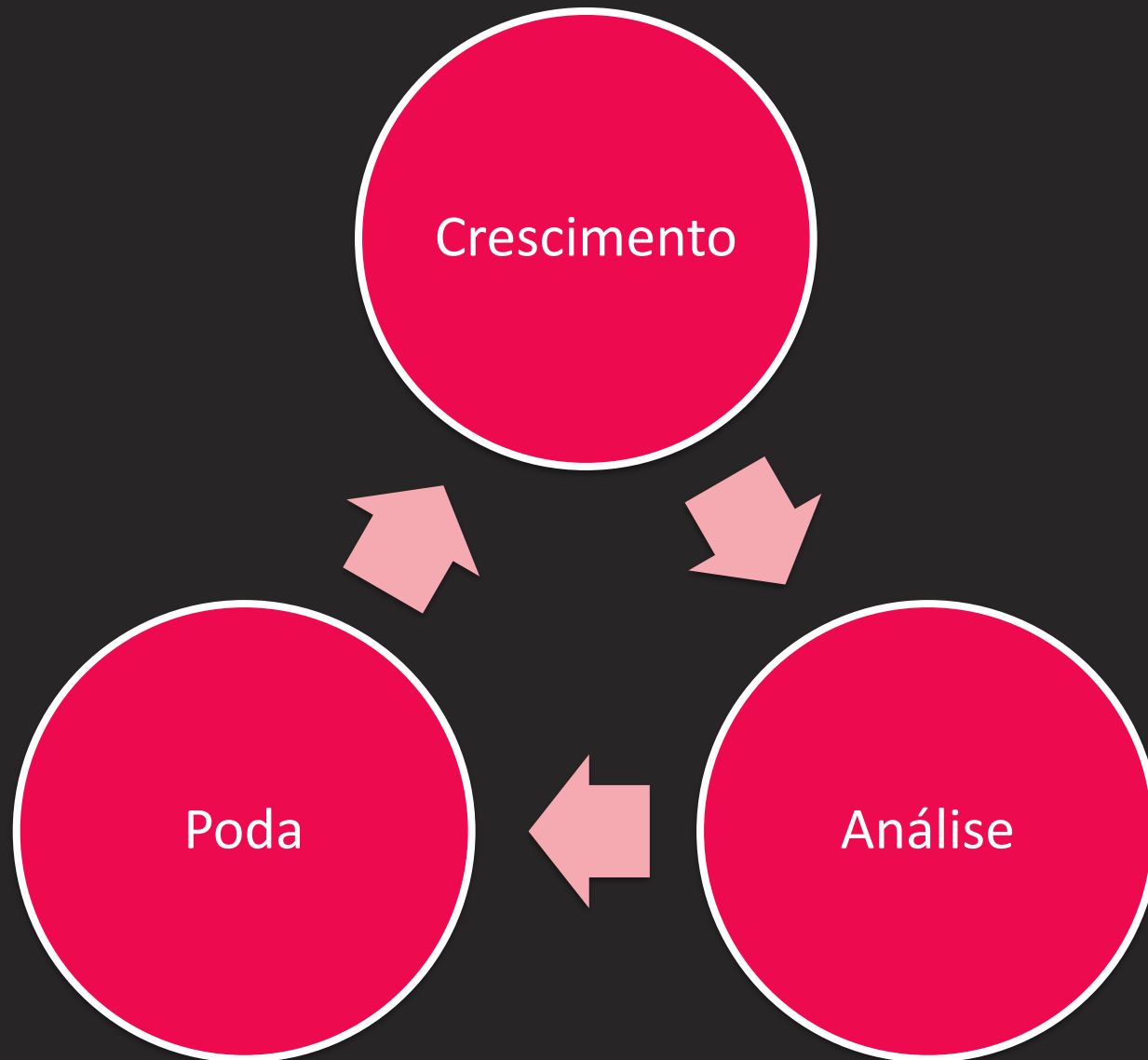
Árvores de decisão



Algoritmos
ID3
C4.5
Cart
...



Processo da árvore de decisão ID3

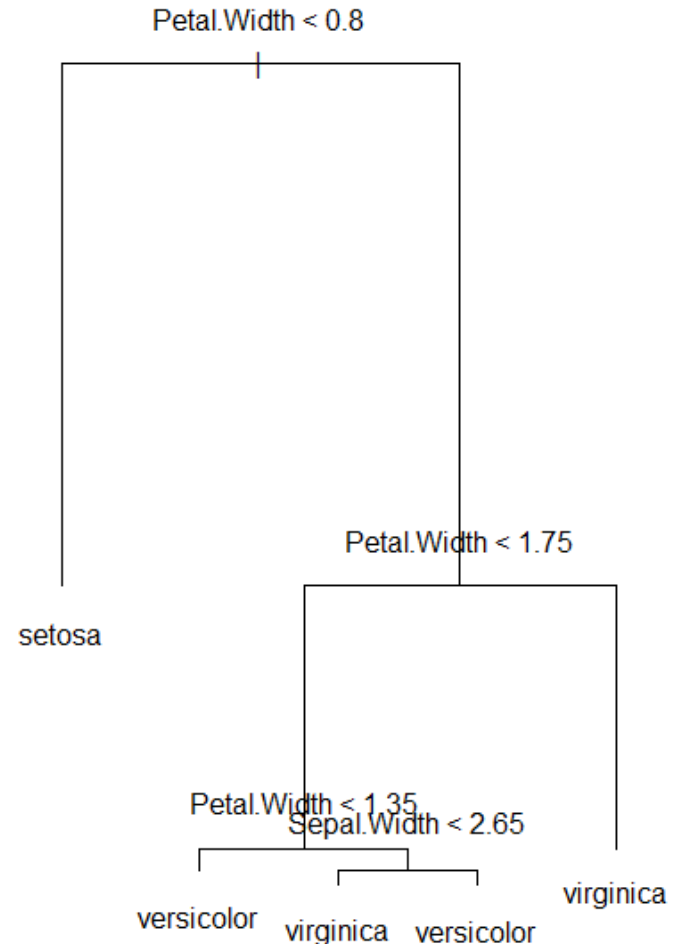


Árvores de decisão para classificação

- `iris.modelo <- tree(data = iris, formula = Species ~ Sepal.Width + Petal.Width)`
- `summary(iris.modelo)`
- `plot(iris.modelo)`
- `text(iris.modelo)`

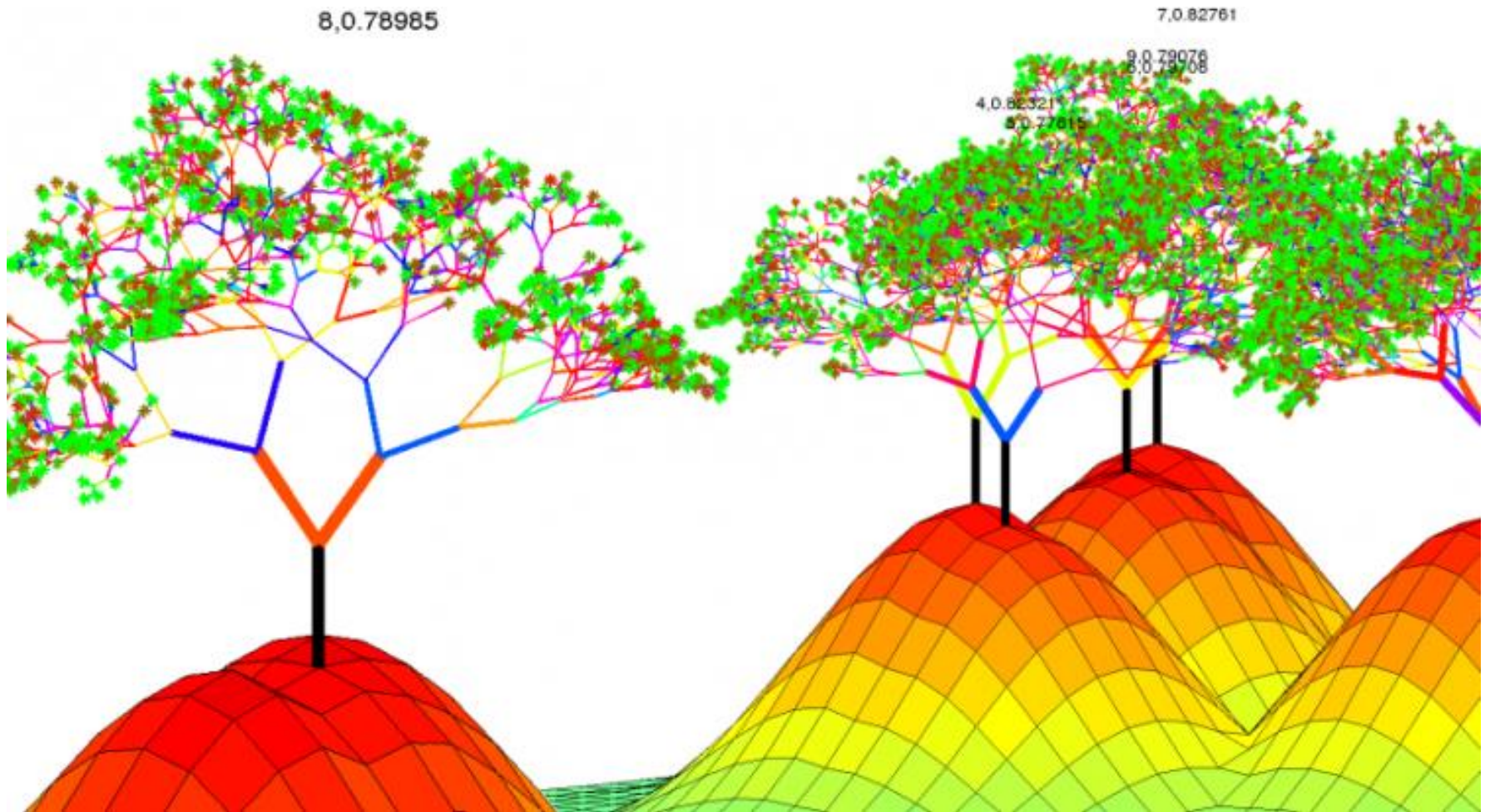
Árvores de decisão serão vistas com mais profundidade nas aulas de estatística.

Nosso objetivo: RANDOM FORESTS



- Fazer uma árvore de decisão para o dataset iris utilizando todos os atributos disponíveis.
 - O resultado melhora ou piora?
 - É um algoritmo determinístico?
 - Precisa de normalização?
 - O que é normalização?

Random Forest



- Ver R Markdown (de classificação)
- Usar o random forest para regressão! Ohhhh
 - Dataset MASS::Boston
 - Campo a descobrir: medv (média de venda)

MBA⁺

Copyright © **2018**

Prof. Elthon Manhas de Freitas

Todos direitos reservados. Reprodução ou divulgação total ou parcial deste documento é expressamente proibido sem o consentimento formal, por escrito, do Professor (autor).