Introdução ao Aprendizado de Máquina

Lucas Gonçalves de Moura Leite

Apresentação

- Lucas Gonçalves de Moura Leite
  - Graduado em Ciência da Computação (UFC 2009-2013)
  - Mestrado em Ciência da Computação (UFC 2014-2016)
    - Lógica - Sistemas Multiagentes
  - Doutorado em andamento Ciência da Computação Computação (UFC 2017-)
    - Aprendizado de Máquina
  - 2016 – 2018
    - Laboratório de Sistemas e Banco de Dados – Aprendizado de máquina para predição de falhas em discos rígidos
  - 2018 – Presente
    - Instituto Atlântico
    - Professor

Aprendizado de Máquina

- Aprendizado de Máquina (*Machine Learning*)
  - "Dotar o computador da capacidade de realizar uma tarefa sem que este tenha sido explicitamente programado para isso"

Aprendizado de Máquina

- Aprendizado de Máquina (*Machine Learning*)
  - "Dotar o computador da capacidade de realizar uma tarefa sem que este tenha sido explicitamente programado para isso"
  - Aprender através da experiência (aprender com os erros)

Aprendizado de Máquina

- Aprendizado de Máquina (*Machine Learning*)
  - "Dotar o computador da capacidade de realizar uma tarefa sem que este tenha sido explicitamente programado para isso"
  - Aprender através da experiência (aprender com os erros)
  - Experiência = Dados (exemplos)

Aprendizado de Máquina

- O estudo de programas que podem aprender a partir de exemplos
- Algoritmos de AM tem a capacidade de generalizar a partir de uma série de exemplos

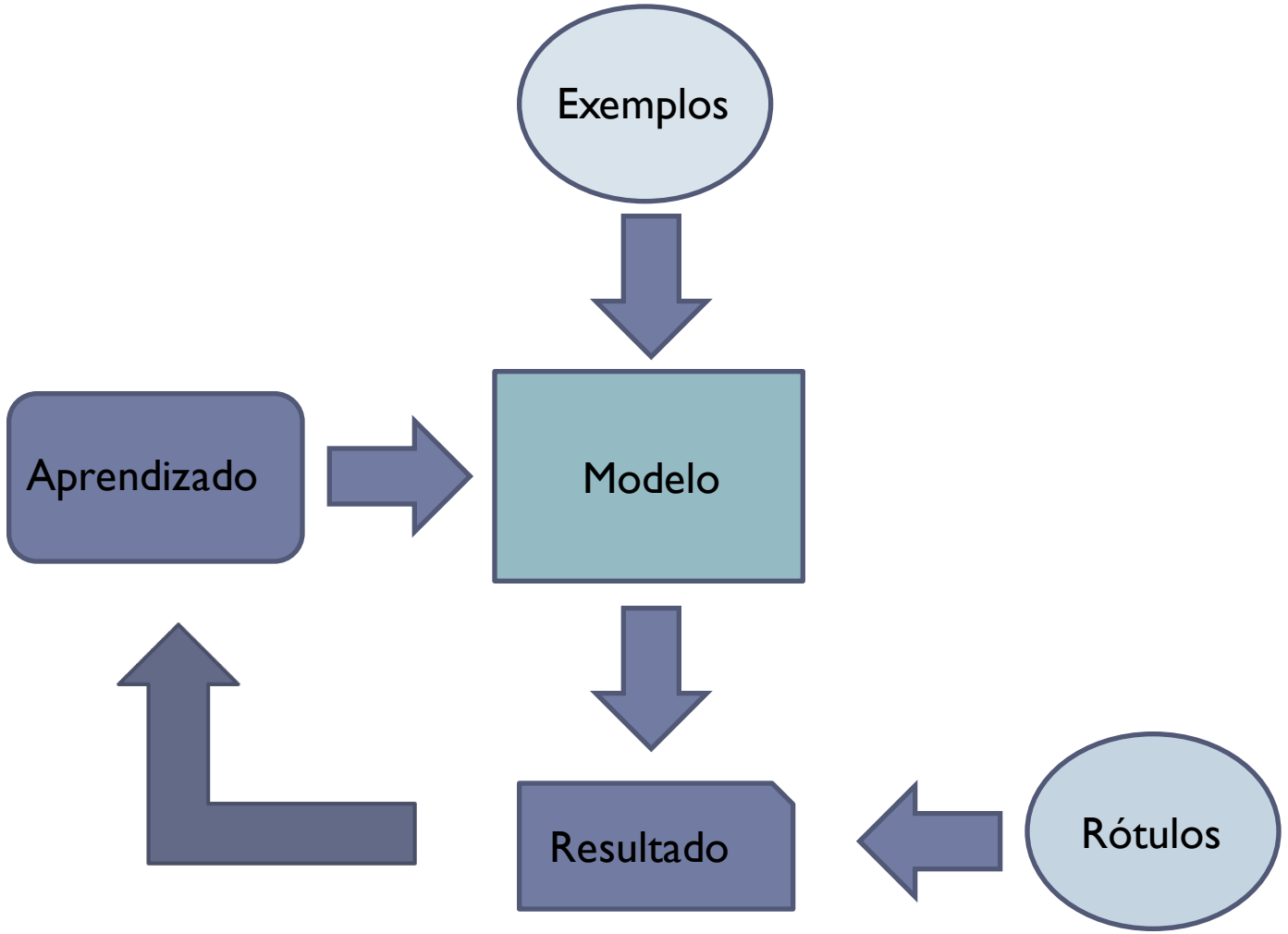
Aprendizado de Máquina

- Realizar tarefas através da estratégia tradicional de fazer código
  - Baseado na utilização de regras
- Tarefas difícieis
  - Reconhecimento de face
  - Reconhecimento de sentenças (*speech to text*)

Aprender a partir de exemplos

- Geração de um conjunto de exemplos rotulados
- Conjunto de treinamento
- Aprender regras e utilizar para exemplos não vistos (capacidade de generalização)

Processo de Treinamento



Aprendizado de Máquina

- Estatística
- Computação
- Otimização

Neste Curso

- Entender os conceitos básicos de AM
- Como aplicar alguns dos algoritmos AM mais populares
- Introdução ao scikit-learn (python)
- Para maior aprofundamento nos conceitos apresentados aqui
  - Machine Learning (coursera.org)

Aprendizado de Máquina

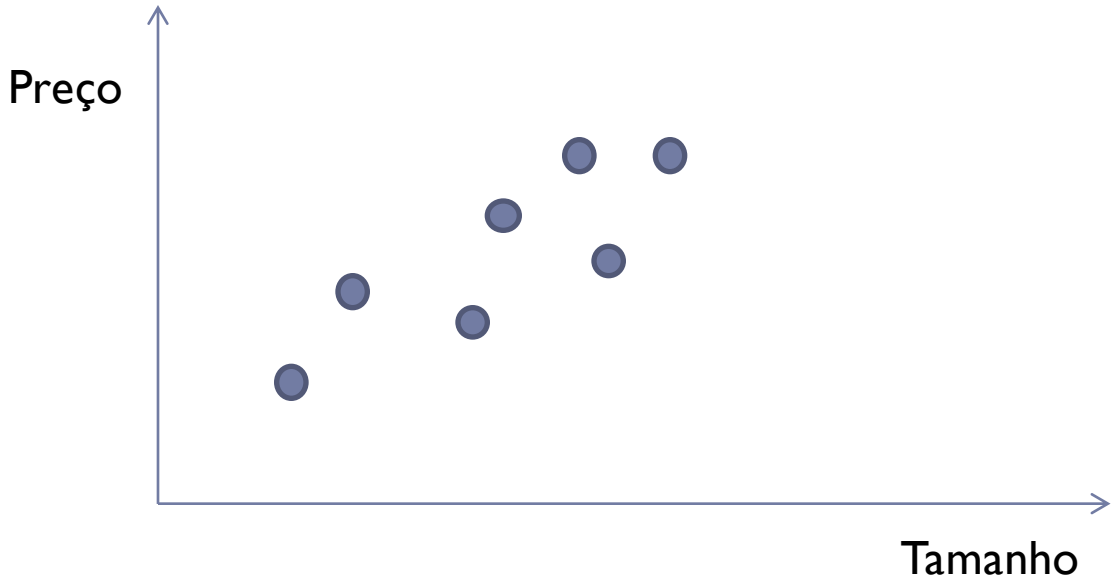
- Paradigmas de Aprendizado
  - Aprendizado Supervisionado
  - Aprendizado Não-Supervisionado

Aprendizado Supervisionado

- Aprender através de dados rotulados

Aprendizado Supervisionado

- Aprender através de dados rotulados
  - Exemplo
    - Preço de imóveis



Aprendizado Supervisionado

- Aprender através de dados rotulados
  - Exemplo
    - Máquina de refrigerante

Aprendizado Não-Supervisionado

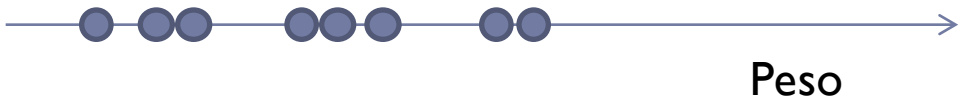
- Aprender através de dados não rotulados

Aprendizado Não-Supervisionado

- Aprender através de dados não rotulados
  - Exemplo

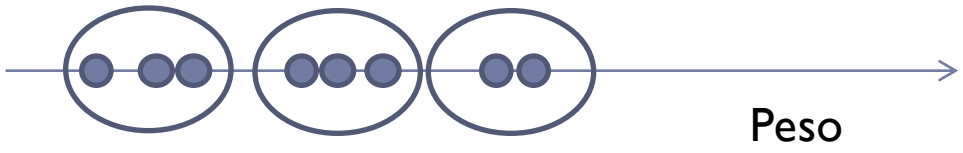
Aprendizado Não-Supervisionado

- Aprender através de dados não rotulados
  - Exemplo



Aprendizado Não-Supervisionado

- Aprender através de dados não rotulados
  - Exemplo



Como aplicar AM

- Representação
  - Escolha dos atributos
  - Escolha do classificador
- Avaliação
  - Critério de avaliação do seu método
- Otimização
  - Encontrar os melhores parâmetros para otimizar o critério de desempenho

Representação

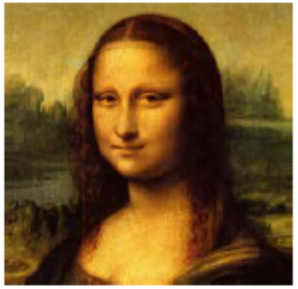
- Extração de atributos (*features*)

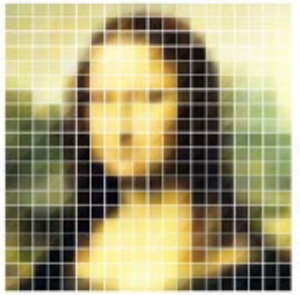
Email

```
To:  Chris Brooks
From:  Daniel Romero
Subject:  Next course offering
Hi Daniel,
Could you please send the outline for the
next course offering?  Thanks! -- Chris
```

Feature	Count
to	1
chris	2
brooks	1
from	1
daniel	2
romero	1
the	2
...	

Imagem

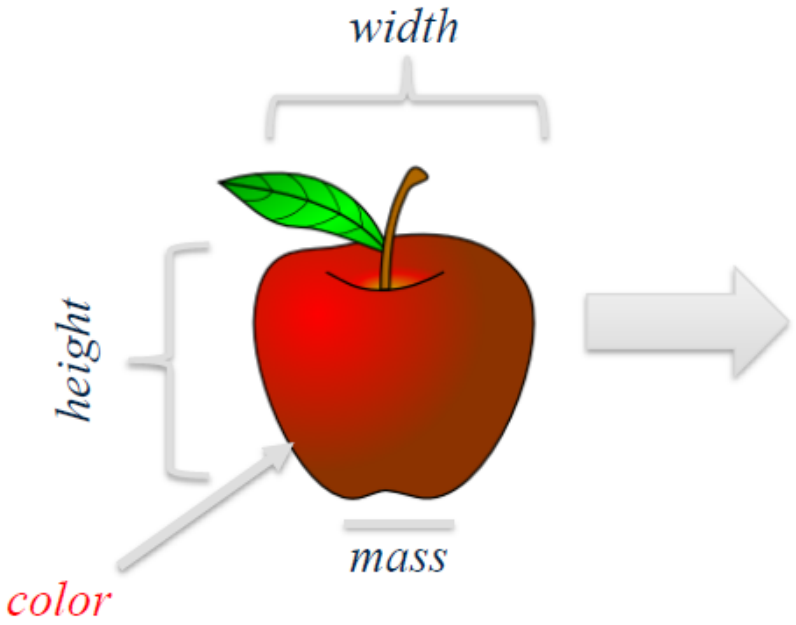




Peixe

Feature	Value
DorsalFin	Yes
MainColor	Orange
Stripes	Yes
StripeColor1	White
StripeColor2	Black
Length	4.3 cm

Atributos



Label information
(available in training data only)

Feature representation

	fruit_label	fruit_name	fruit_subtype	mass	width	height	color_score
18	1	apple	cripps_pink	162	7.5	7.1	0.83

Representação

- Escolha do método de AM
  - Muitas opções
  - Critérios
    - Complexidade
    - Interpretabilidade
    - Custo computacional

Avaliação

- Como verificar a qualidade do seu modelo
  - Vários Critérios
    - Natureza do tarefa (classificação, regressão ...)
    - Características dos dados
    - Natureza da aplicação

Otimização

- Performance do modelo depende das escolhas da etapa de representação
  - Atributos
    - Quantos e quais
  - Método de AM
    - Natureza e parâmetros

AM com Python

- Distribuição Anaconda
  - Scikit Learn
  - Pandas
  - Numpy
  - SciPy
  - Matplotlib

Scikit Learn

scikit-learn: Python Machine Learning Library

- scikit-learn Homepage
  http://scikit-learn.org/
- scikit-learn User Guide
  http://scikit-learn.org/stable/user_guide.html
- scikit-learn API reference
  http://scikit-learn.org/stable/modules/classes.html
- In Python, we typically import classes and functions we need like this:

```
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
```

SciPy

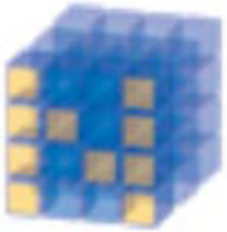
SciPy Library: Scientific Computing Tools

http://www.scipy.org/

- Provides a variety of useful scientific computing tools, including statistical distributions, optimization of functions, linear algebra, and a variety of specialized mathematical functions.
- With scikit-learn, it provides support for *sparse matrices*, a way to store large tables that consist mostly of zeros.
- Example import: `import scipy as sp`

Numpy

NumPy: Scientific Computing Library



http://www.numpy.org/

- Provides fundamental data structures used by scikit-learn, particularly multi-dimensional arrays.
- Typically, data that is input to scikit-learn will be in the form of a NumPy array.
- Example import: `import numpy as np`

Pandas

Pandas: Data Manipulation and

S http://pandas.pydata.org/

- Provides key data structures like `DataFrame`
- Also, support for reading/writing data in different formats
- Example import: `import pandas as pd`

Matplotlib

matplotlib and other plotting libraries

http://matplotlib.org/

- We typically use matplotlib's **pyplot** module:

```
import matplotlib.pyplot as plt
```

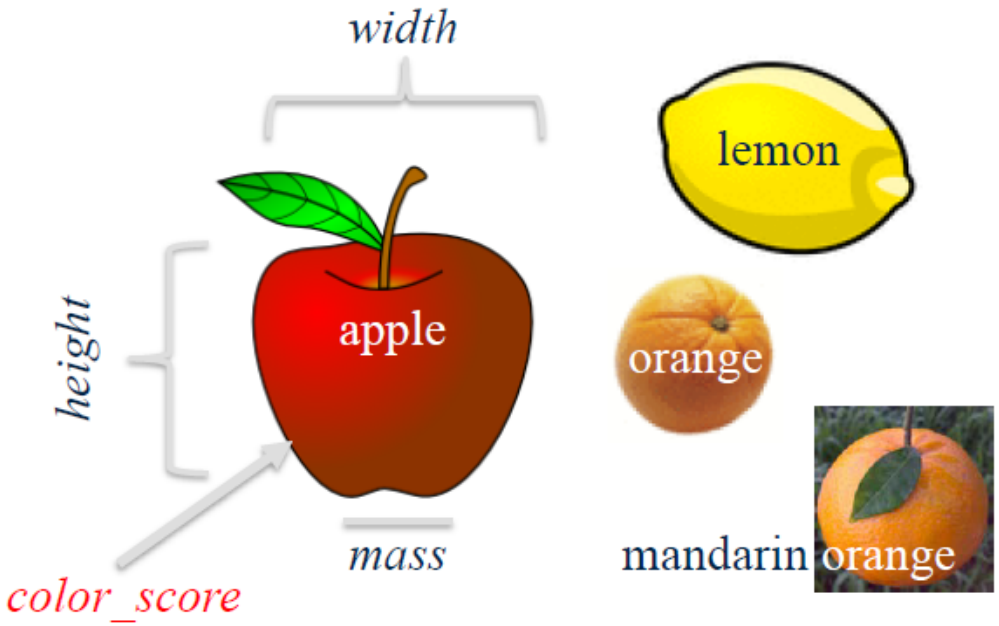
- We also sometimes use the **seaborn** visualization library (http://seaborn.pydata.org/)

```
import seaborn as sn
```

- And sometimes the **graphviz** plotting library:

```
import graphviz
```

Uma Aplicação Simples



	fruit_label	fruit_name	fruit_subtype	mass	width	height	color_score
0	1	apple	granny_smith	192	8.4	7.3	0.55
1	1	apple	granny_smith	180	8.0	6.8	0.59
2	1	apple	granny_smith	176	7.4	7.2	0.60
3	2	mandarin	mandarin	86	6.2	4.7	0.80
4	2	mandarin	mandarin	84	6.0	4.6	0.79
5	2	mandarin	mandarin	80	5.8	4.3	0.77
6	2	mandarin	mandarin	80	5.9	4.3	0.81
7	2	mandarin	mandarin	76	5.8	4.0	0.81
8	1	apple	braeburn	178	7.1	7.8	0.92
9	1	apple	braeburn	172	7.4	7.0	0.89
10	1	apple	braeburn	166	6.9	7.3	0.93
11	1	apple	braeburn	172	7.1	7.6	0.92
12	1	apple	braeburn	154	7.0	7.1	0.88
13	1	apple	golden_delicious	164	7.3	7.7	0.70
14	1	apple	golden_delicious	152	7.6	7.3	0.69
15	1	apple	golden_delicious	156	7.7	7.1	0.69
16	1	apple	golden_delicious	156	7.6	7.5	0.67

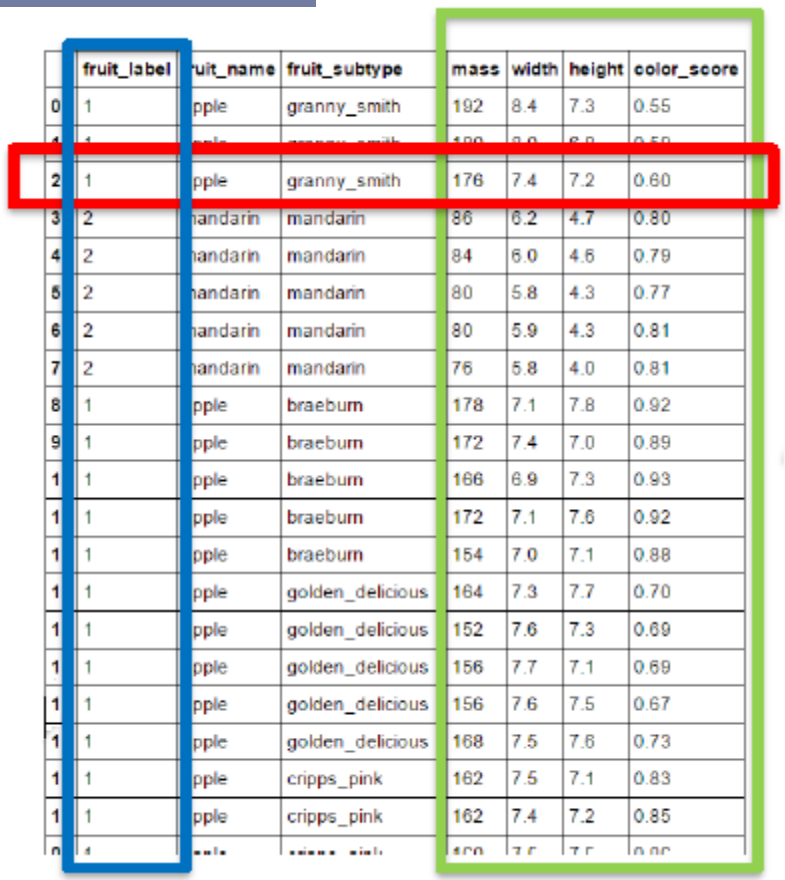
`fruit_data_with_colors.txt`

Dados

Rótulo ou Label

Atributos ou features

Exemplo ou instância



	fruit_label	fruit_name	fruit_subtype	mass	width	height	color_score
0	1	pple	granny_smith	192	8.4	7.3	0.55
2	1	pple	granny_smith	176	7.4	7.2	0.60
3	2	nandarin	mandarin	86	6.2	4.7	0.80
4	2	nandarin	mandarin	84	6.0	4.6	0.79
5	2	nandarin	mandarin	80	5.8	4.3	0.77
6	2	nandarin	mandarin	80	5.9	4.3	0.81
7	2	nandarin	mandarin	76	5.8	4.0	0.81
8	1	pple	braeburn	178	7.1	7.8	0.92
9	1	pple	braeburn	172	7.4	7.0	0.89
1	1	pple	braeburn	166	6.9	7.3	0.93
1	1	pple	braeburn	172	7.1	7.6	0.92
1	1	pple	braeburn	154	7.0	7.1	0.88
1	1	pple	golden_delicious	164	7.3	7.7	0.70
1	1	pple	golden_delicious	152	7.6	7.3	0.69
1	1	pple	golden_delicious	156	7.7	7.1	0.69
1	1	pple	golden_delicious	156	7.6	7.5	0.67
1	1	pple	golden_delicious	168	7.5	7.6	0.73
1	1	pple	cripps_pink	162	7.5	7.1	0.83
1	1	pple	cripps_pink	162	7.4	7.2	0.85

Carregando os dados

```
%matplotlib notebook
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
from sklearn.model_selection import train_test_split

fruits = pd.read_table('fruit_data_with_colors.txt')
```

```
fruits.shape
```

```
(59, 7)
```

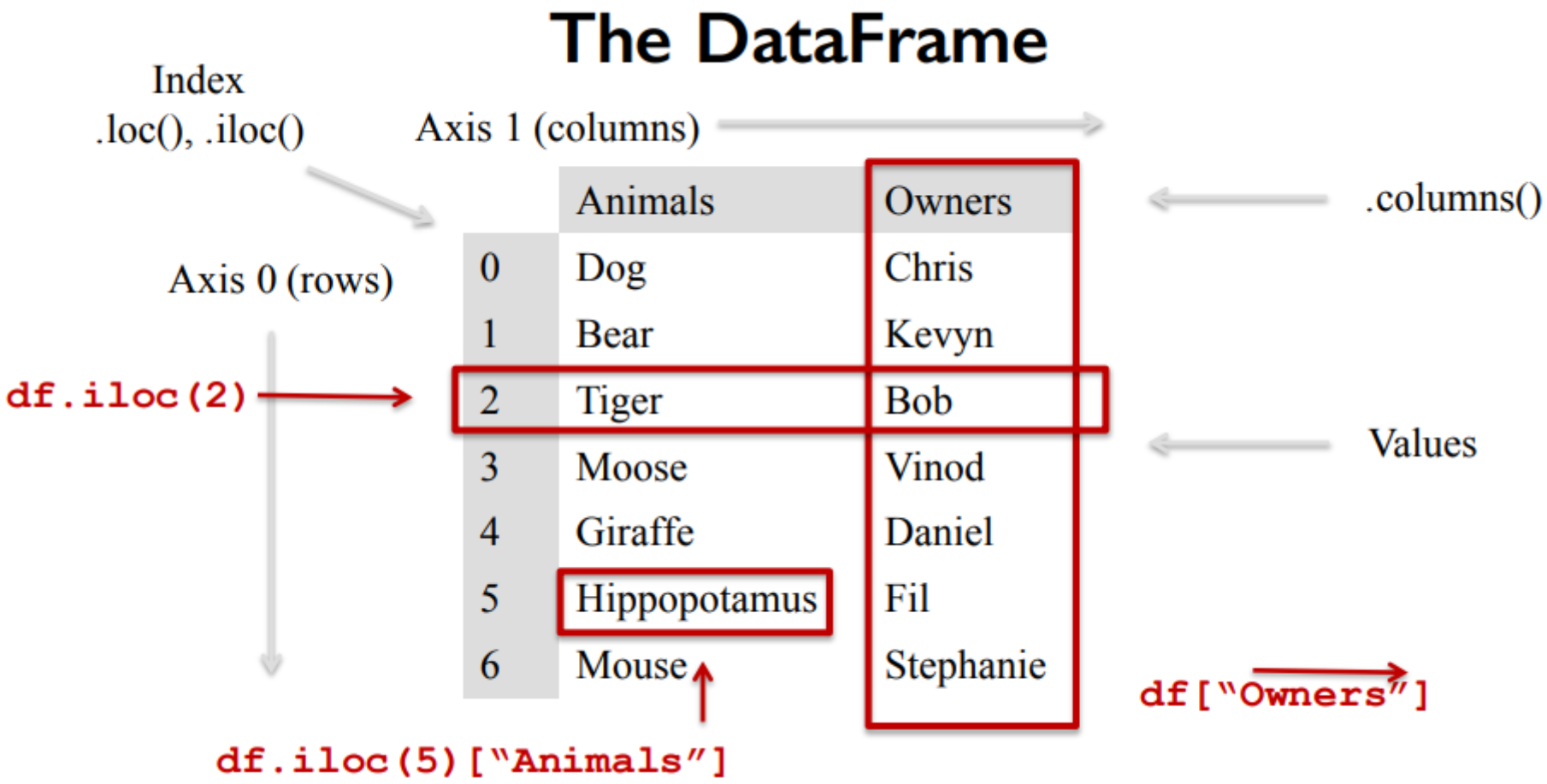
```
fruits.head(2)
```

	fruit_label	fruit_name	fruit_subtype	mass	width	height	color_score
0	1	apple	granny_smith	192	8.4	7.3	0.55
1	1	apple	granny_smith	180	8.0	6.8	0.59

Estruturas de Dados

- Python
  - Listas (Vetores e Matrizes)
  - Dicionário
- Pandas
  - Series
    - Lista e dicionário
    - Itens ordenados e existem labels que podem ser usados para fazer consultas
  - DataFrame (.DataFrame)
    - Multidimensional
    - Itens ordenados
    - Possibilita consulta em linhas e colunas

DataFrame


The DataFrame
Index
.loc(), .iloc()
Axis 1 (columns)
.columns()
Axis 0 (rows)
df.iloc(2)
Values
df["Owners"]
df.iloc(5)["Animals"]
Animals
Owners
0
Dog
Chris
1
Bear
Kevyn
2
Tiger
Bob
3
Moose
Vinod
4
Giraffe
Daniel
5
Hippopotamus
Fil
6
Mouse
Stephanie

Selecionando linhas e colunas

	fruit_label	fruit_name	fruit_subtype	mass	width	height	color_score
0	1	apple	granny_smith	192	8.4	7.3	0.55
1	1	apple	granny_smith	180	8.0	6.8	0.59
2	1	apple	granny_smith	176	7.4	7.2	0.60
3	2	mandarin	mandarin	86	6.2	4.7	0.80
4	2	mandarin	mandarin	84	6.0	4.6	0.79

```
fruits['mass']
fruits[:3]
fruits['mass'][:3]
freq = fruits['fruit_name'].value_counts()
freq.plot(kind = 'bar')
plt.show()
```

Coluna 'mass'

3 primeiras linhas

3 primeiras linhas da coluna 'mass'

Contagem de ocorrências do atributo 'fruit_name'

Selecionando linhas e colunas

```
macas =  fruits['fruit_name'] == 'apple'
fruits[fruits['fruit_name'] == 'apple']
pesadas = fruits['mass'] > 175
fruits[macas & pesadas]
```

Indices das maçãs

Instancias cujo atributo 'fruit_name' é maçã

Indices das frutas que pesam mais que 175

Maçãs que pesam mais que 175

Plotando os dados

```
X1 = fruits['width']
X2 = fruits['height']
plt.scatter(X1, X2, c = y)
plt.show()
pd.scatter_matrix(X, c= y, figsize=(9,9))
```

Treino x Teste

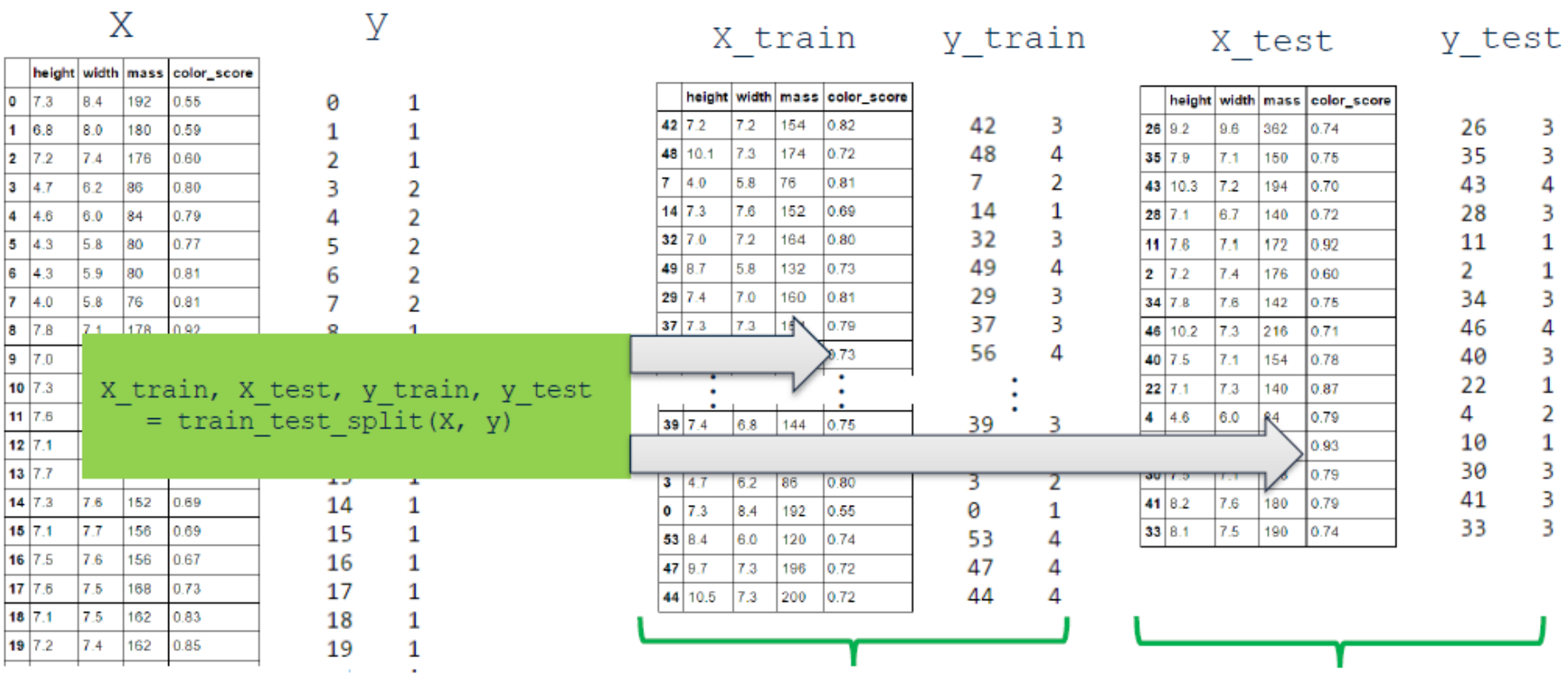
- Aprendizado de máquina
  - Capacidade de generalização
  - Como verificar ?

Treino x Teste

```
# plotting a scatter matrix
from matplotlib import cm

X = fruits[['height', 'width', 'mass', 'color_score']]
y = fruits['fruit_label']
X_train, X_test, y_train, y_test = train_test_split(X, y, random_state=0)
```

Treino x Teste



Dado Original

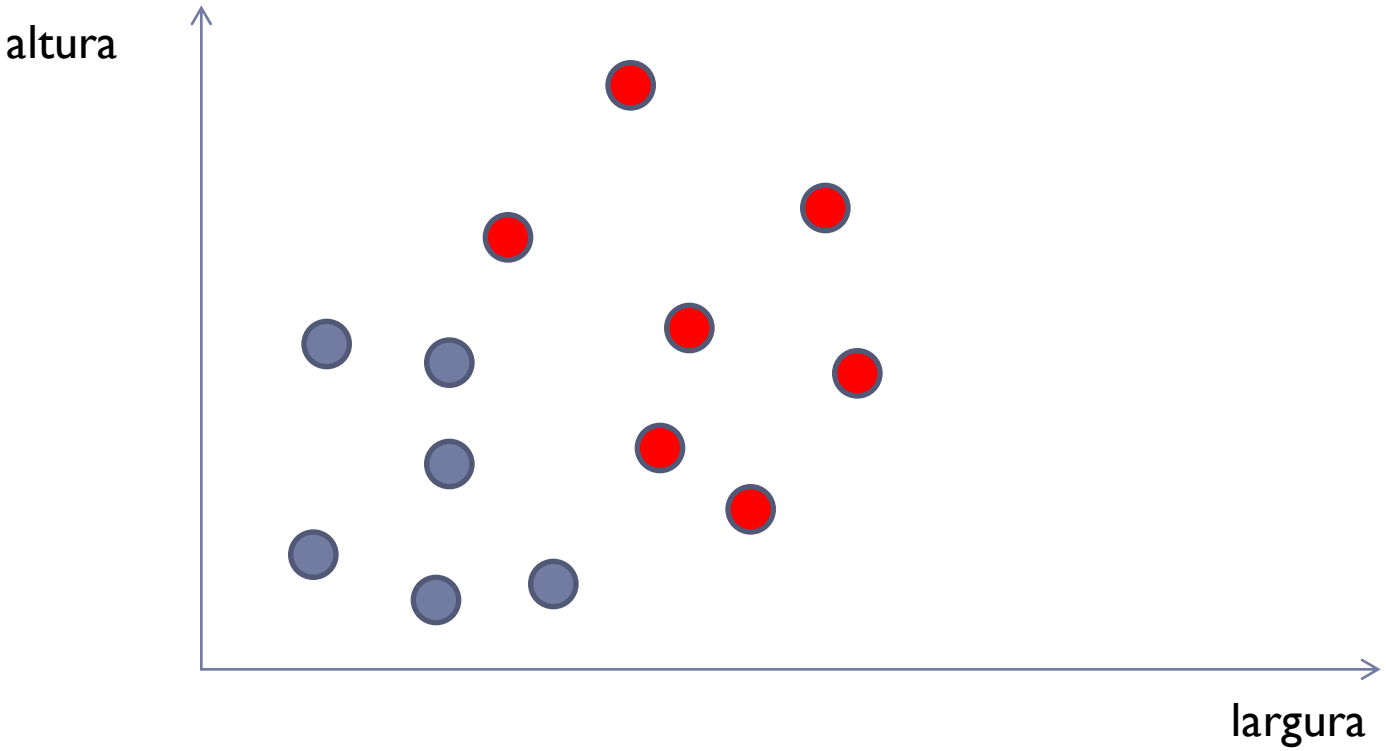
Conjunto de Treino

Conjunto de Teste

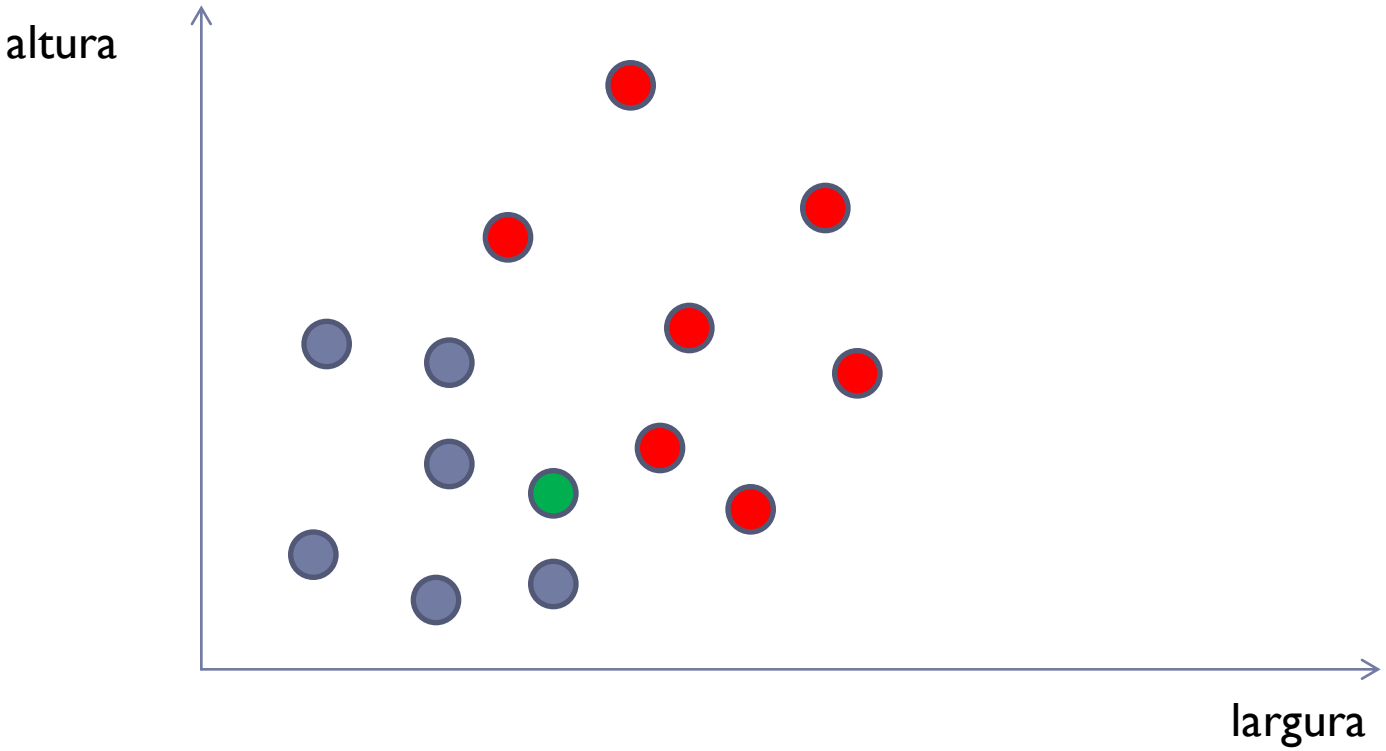
Primeiro Classificador (K-NN)

- Para um novo dado, verifica-se quais dados do conjunto de treinamento são mais semelhantes.
  - Atributos (x)
- A classe escolhida é a classe da maioria dos dados mais próximos

K-NN



K-NN



K-NN - Classificação

- Armazena n exemplos de cada classe (conjunto de treinamento)
- Calcula a distância do novo ponto para todos o n exemplos de cada classe

K-NN - Classificação

- Armazena n exemplos de cada classe (conjunto de treinamento)
- Calcula a distância do novo ponto para todos o n exemplos de cada classe
- Verifica quais são os k pontos mais próximos

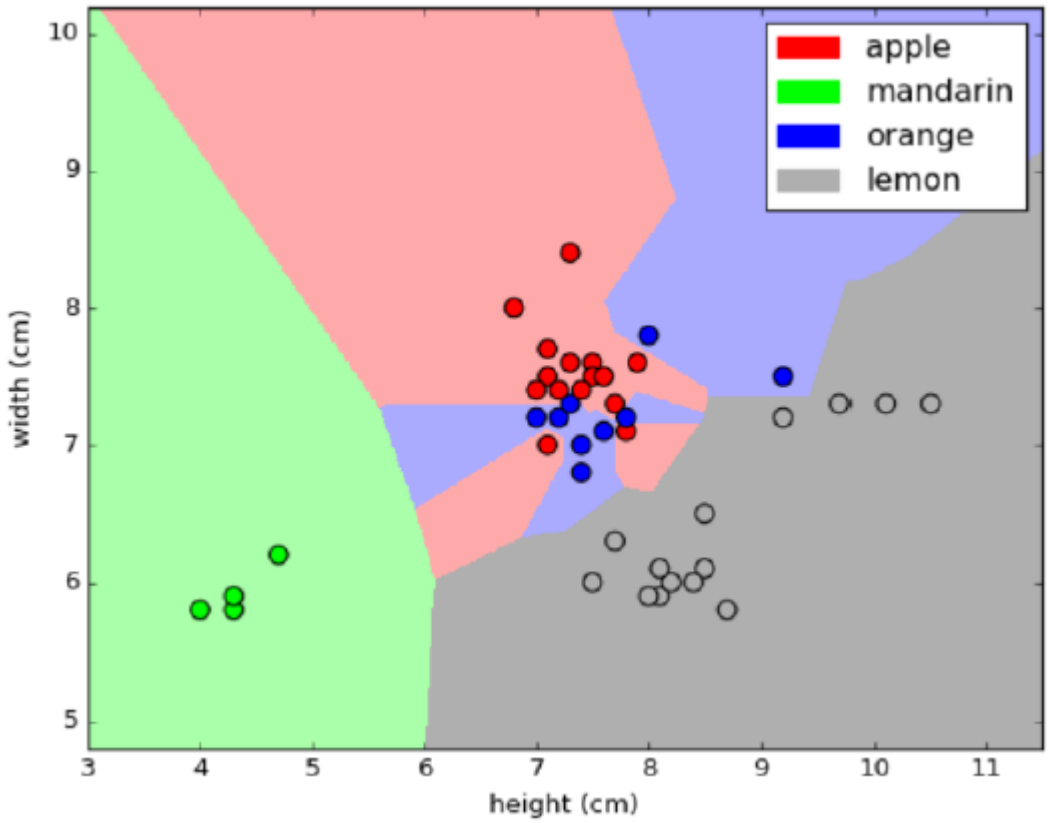
K-NN - Classificação

- Armazena n exemplos de cada classe (conjunto de treinamento)
- Calcula a distância do novo ponto para todos o n exemplos de cada classe
- Verifica quais são os k pontos mais próximos
- Verifica a classe destes k pontos

K-NN - Classificação

- Armazena n exemplos de cada classe (conjunto de treinamento)
- Calcula a distância do novo ponto para todos o n exemplos de cada classe
- Verifica quais são os k pontos mais próximos
- Verifica a classe destes k pontos
- A classe do novo dado será igual a classe da maioria

K-NN para as frutas (K=1)



K-NN no Scikit Learn

```
from sklearn.neighbors import KNeighborsClassifier

knn = KNeighborsClassifier(n_neighbors = 5)
```

```
knn.fit(X_train, y_train)
```

```
knn.score(X_test, y_test)
```

```
# first example: a small fruit with mass 20g, width 4.3 cm, height 5.5 cm
fruit_prediction = knn.predict([[20, 4.3, 5.5]])
print(fruit_prediction)
```

```
# first example: a small fruit with mass 20g, width 4.3 cm, height 5.5 cm
fruit_prediction = knn.predict([[20, 4.3, 5.5]])
print(fruit_prediction)

fruit_prediction = knn.predict(X_test.iloc[0])
print(fruit_prediction)

y_test.iloc[0]
#lookup_fruit_name[fruit_prediction[0]]
```

Superfície de Classificação

```
from adspy_shared_utilities import plot_fruit_knn

plot_fruit_knn(X_train, y_train, 5, 'uniform')   # we choose 5 nearest neighbors
```

- Variem o valor de k

Efeito de K na acurácia

```
k_range = range(1,20)
scores = []

for k in k_range:
    knn = KNeighborsClassifier(n_neighbors = k)
    knn.fit(X_train, y_train)
    scores.append(knn.score(X_test, y_test))

plt.figure()
plt.xlabel('k')
plt.ylabel('accuracy')
plt.scatter(k_range, scores)
plt.xticks([0,5,10,15,20]);
```

Exercício

- Carregar os dados do dataset iris
- Visualizar os dados (escolher dois atributos)
- Fazer o scatterplot
- Classificar usando K-NN (variar distancia e k)
- Visualizar as superfícies de decisão

```
import numpy as np
import pandas as pd
from sklearn.datasets import load_iris
```

Exercício

- Carregar os dados do dataset iris
- Visualizar os dados (escolher dois atributos)
- Fazer o scatterplot
- Classificar usando K-NN (variar distancia e k)
- Visualizar as superfícies de decisão

```
import numpy as np
import pandas as pd
from sklearn.datasets import load_iris

data1 = pd.DataFrame(data= np.c_[iris['data'], iris['target']],
                     columns= iris['feature_names'] + ['target'])
```

K-NN

- Vantagens
  - Simples
  - Treinamento rápido
- Desvantagens
  - Custo computacional do teste
  - Influenciado pela ordem de grandeza dos atributos