Aprendizado de Máquina - Exemplo do K-NN

1 - Importar os módulos necessários e carregar a base de dados de frutas

In [1]:

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

In [4]:

```
fruits = pd.read_table('./Data/fruit_data_with_colors.txt')
fruits.head()
```

Out[4]:

	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

2 - Mapeamento para facilitar a interpretação dos resultados

In [5]:

```
# criar um mapeamento do valor do rótulo da fruta para o nome da fruta para facilitar a i
lookup_fruit_name = dict(zip(fruits.fruit_label.unique(), fruits.fruit_name.unique() ))
lookup_fruit_name
```

Out[5]:

```
{1: 'apple', 2: 'mandarin', 3: 'orange', 4: 'lemon'}
```

3 - Separando conjuto de treinamento de conjunto de testes

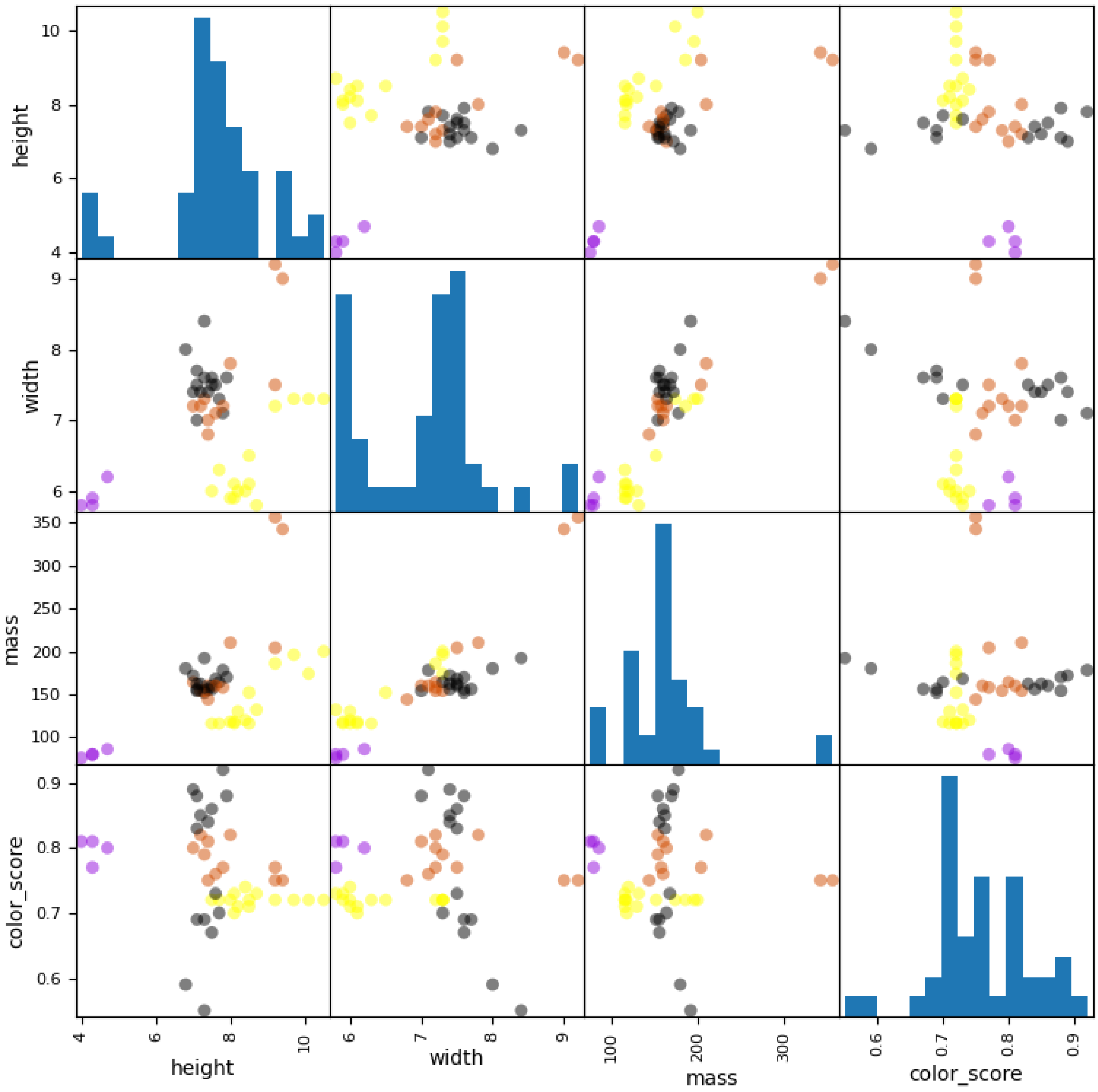
In [6]:

```
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)
```

4 - Examinando inicialmente os dados com uma matriz de dispersão

In [9]:

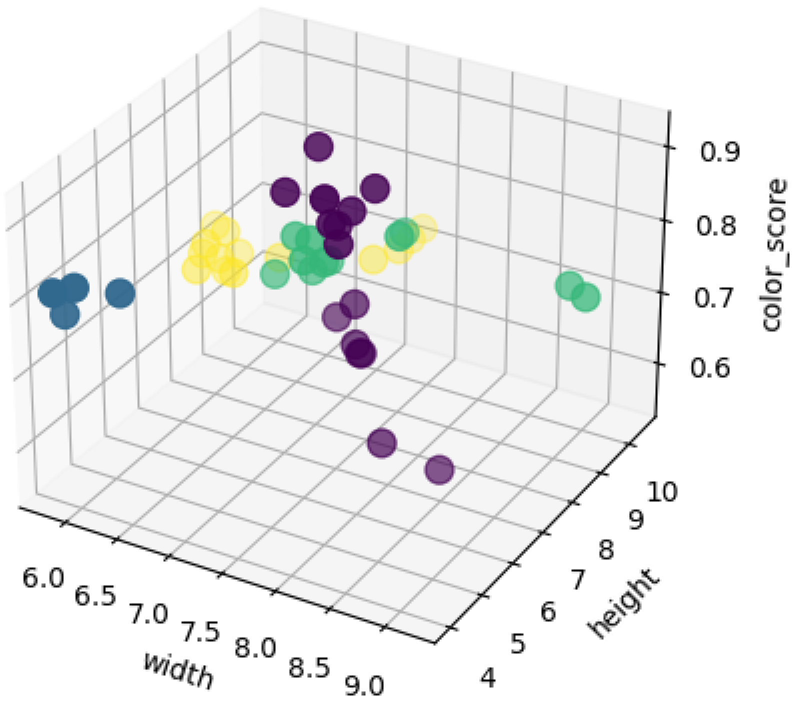
```
# plotando uma matriz de dispersão ou gráfico de pares de características
from matplotlib import cm

cmap = cm.get_cmap('gnuplot')
scatter = pd.plotting.scatter_matrix(X_train,
                                     c=y_train,
                                     marker='o',
                                     s=40,
                                     hist_kwds={'bins':15},
                                     figsize=(9,9), cmap=cmap )
```



5 - Gráfico de dispersão 3D

In [8]:

```
from mpl_toolkits.mplot3d import Axes3D

fig = plt.figure()
ax = fig.add_subplot(111, projection = '3d')
ax.scatter(X_train['width'], X_train['height'], X_train['color_score'], c = y_train, marke
ax.set_xlabel('width')
ax.set_ylabel('height')
ax.set_zlabel('color_score')
plt.show()
```



6 - Separar (novamente) o conjunto de treinamento do conjunto de testes

In [10]:

```
X = fruits[['height', 'width', 'mass']]
y = fruits['fruit_label']

X_train, X_test, y_train, y_test = train_test_split(X, y, random_state=0)

# 75% / 25% padrão
```

7 - Cria o objeto Classificador

In [11]:

```
from sklearn.neighbors import KNeighborsClassifier

knn = KNeighborsClassifier(n_neighbors = 5)
```

8 - Treina o classificador usando o conjunto de dados de treinamento

In [13]:

```
knn.fit(X_train, y_train)

print(len(X_train))
print(len(X_test))
```

```
44
15
```

9 - Estimar a precisão do classificador em dados futuros, usando os dados de teste

In [14]:

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

```
0.5333333333333333
```

Out[14]:

10 - Usar o classificador k-NN já treinado para classificar objetos inéditos

In [17]:

```
# fruta com massa 20g, largura 4.3 cm e altura 5.5 cm
fruit_prediction = knn.predict([[20, 4.3, 5.5], [100, 6.3, 8.5]])

print(fruit_prediction)
#lookup_fruit_name[fruit_prediction]
```

```
[2 2]
/Users/marinaramalhetedesouza/opt/anaconda3/envs/ml-impa/lib/python3.8/site-packages/sklea
rn/base.py:445: UserWarning: X does not have valid feature names, but KNeighborsClassifier
was fitted with feature names
  warnings.warn(
```

In [18]:

```
knn.predict?
```

11 - Plotar os limites de decisão do classificador k-NN

In [19]:

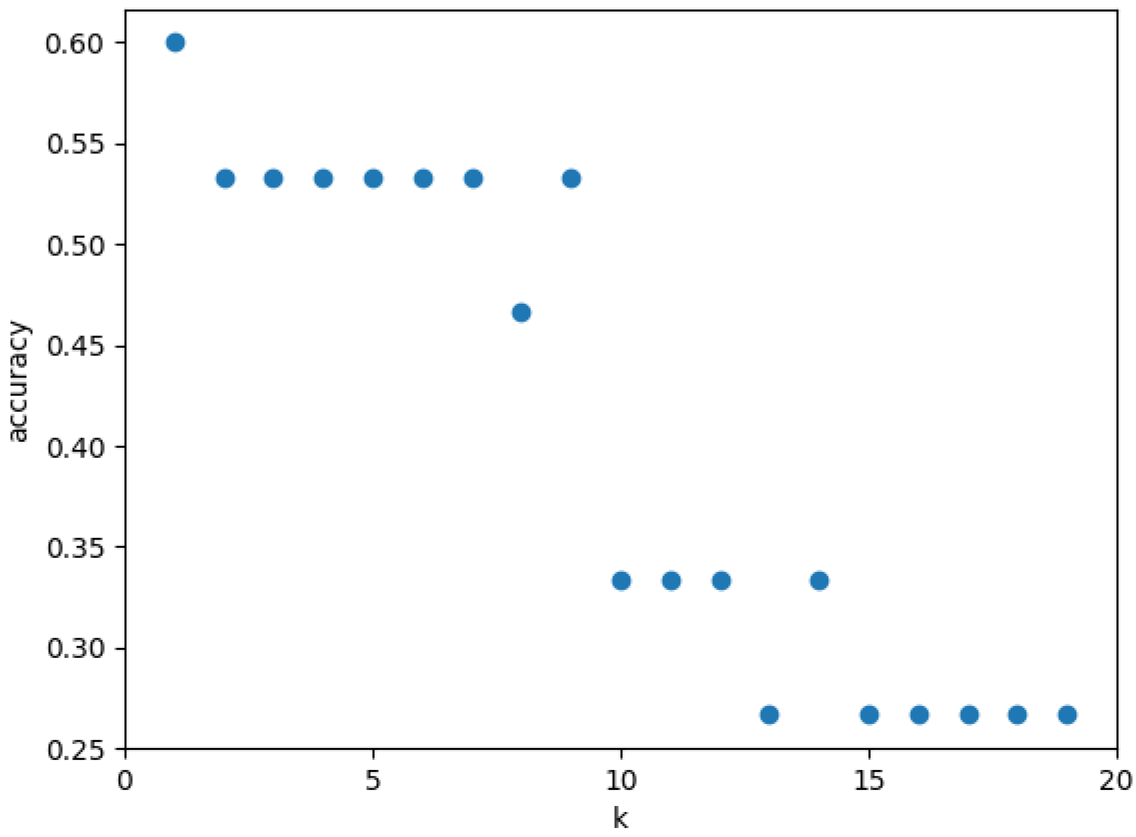
```
from sklearn import neighbors
from matplotlib.colors import ListedColormap, BoundaryNorm
import matplotlib.patches as mpatches
import graphviz
from sklearn.tree import export_graphviz
import matplotlib.patches as mpatches

def plot_fruit_knn(X, y, n_neighbors, weights):
    X_mat = X[['height', 'width']].as_matrix()
    y_mat = y.as_matrix()

    # Mapas de cor
    cmap_light = ListedColormap(['#FFAAAA', '#AAFFAA', '#AAAAFF','#AFAFAF'])
    cmap_bold  = ListedColormap(['#FF0000', '#00FF00', '#0000FF','#AFAFAF'])

    clf = neighbors.KNeighborsClassifier(n_neighbors, weights=weights)
    clf.fit(X_mat, y_mat)

    # Atribuir uma cor no mapa de cores a cada ponto da malha

    mesh_step_size = .01
    plot_symbol_size = 50

    x_min, x_max = X_mat[:, 0].min() - 1, X_mat[:, 0].max() + 1
    y_min, y_max = X_mat[:, 1].min() - 1, X_mat[:, 1].max() + 1
    xx, yy = np.meshgrid(np.arange(x_min, x_max, mesh_step_size),
                         np.arange(y_min, y_max, mesh_step_size))
    Z = clf.predict(np.c_[xx.ravel(), yy.ravel()])

    Z = Z.reshape(xx.shape)
    plt.figure()
    plt.pcolormesh(xx, yy, Z, cmap=cmap_light)

    # Plotar os pontos de treinamento
    plt.scatter(X_mat[:, 0], X_mat[:, 1], s=plot_symbol_size, c=y, cmap=cmap_bold, edgecol
    plt.xlim(xx.min(), xx.max())
    plt.ylim(yy.min(), yy.max())

    patch0 = mpatches.Patch(color='#FF0000', label='apple')
    patch1 = mpatches.Patch(color='#00FF00', label='mandarin')
    patch2 = mpatches.Patch(color='#0000FF', label='orange')
    patch3 = mpatches.Patch(color='#AFAFAF', label='lemon')
    plt.legend(handles=[patch0, patch1, patch2, patch3])


    plt.xlabel('height (cm)')
    plt.ylabel('width (cm)')

    plt.show()


plot_fruit_knn(X_train, y_train, 5, 'uniform')   # 5 vizinhos
```

```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
/var/folders/01/_r7b02r11p15j0s54gb9x0040000gn/T/ipykernel_22501/2012486047.py in <module>
     50 
     51 
---> 52 plot_fruit_knn(X_train, y_train, 5, 'uniform')   # 5 vizinhos

/var/folders/01/_r7b02r11p15j0s54gb9x0040000gn/T/ipykernel_22501/2012486047.py in plot_fru
it_knn(X, y, n_neighbors, weights)
      7 
      8 def plot_fruit_knn(X, y, n_neighbors, weights):
----> 9     X_mat = X[['height', 'width']].as_matrix()
     10     y_mat = y.as_matrix()
     11 

~/opt/anaconda3/envs/ml-impa/lib/python3.8/site-packages/pandas/core/generic.py in __getat
tr__(self, name)
   5485         ):
   5486             return self[name]
-> 5487         return object.__getattribute__(self, name)
   5488 
   5489     def __setattr__(self, name: str, value) -> None:

AttributeError: 'DataFrame' object has no attribute 'as_matrix'
```

12 - Quão sensível é a precisão da classificação k-NN para a escolha do parâmetro 'k'?

In [20]:

```
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]);
```



```
In [ ]:
```