Clasificación de Frutas

****Elemento de estudio:****

 En este trabajo se tomo en cuenta la clasificación de 3 frutas:

* Mante
* Naranja
* Manzana

**Razón:**

Todas de ellas son consideradas frutas y poseen las propiedades de tener un alto, un ancho y una circunferencia en cierto modo.

****Factores de clasificación:****

* Alto (cm)
* Ancho (cm)
* Circunferencia (cm)
* Radio Circunferencia (cm)

Para el calculo del radio de la circunferencia se tomo en cuenta la formula:

R = Perimetro/(2\*PI)

Datos:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Alto | Ancho | Circunferencia | Radio | Etiqueta |
| 11 | 7 | 17 | 2.7 | 0 |
| 10 | 6.5 | 19 | 3.0 | 0 |
| 8 | 6 | 17.5 | 2.8 | 0 |
| 9 | 7 | 17.5 | 2.8 | 0 |
| 8.3 | 5 | 16 | 2.5 | 0 |
| 7.5 | 6 | 17.2 | 2.7 | 0 |
| 11.2 | 6.7 | 19.5 | 3.1 | 0 |
| 11.5 | 6 | 17.8 | 2.8 | 0 |
| 11.1 | 5.5 | 16.9 | 2.7 | 0 |
| 9 | 6 | 17.8 | 2.8 | 0 |
| 11.3 | 5.5 | 18 | 2.9 | 0 |
| 10.4 | 5.5 | 15.5 | 2.5 | 0 |
| 10.9 | 7.5 | 17.3 | 2.8 | 0 |
| 9 | 6 | 18.5 | 2.9 | 0 |
| 8.2 | 6.3 | 17.6 | 2.8 | 0 |
| 8.9 | 7.1 | 17.3 | 2.8 | 0 |
| 8.5 | 4.9 | 16.3 | 2.6 | 0 |
| 11 | 6.5 | 19 | 3.0 | 0 |
| 10.9 | 5.3 | 16.5 | 2.6 | 0 |
| 8.9 | 5.9 | 17.4 | 2.8 | 0 |
| 11.2 | 5.4 | 17 | 2.7 | 0 |
| 10.2 | 5.2 | 15.6 | 2.5 | 0 |
| 8.1 | 5.4 | 13.5 | 2.1 | 0 |
| 10.4 | 5.5 | 15.5 | 2.5 | 0 |
| 10.9 | 7.5 | 17.3 | 2.8 | 0 |
| 9 | 6 | 18.5 | 2.9 | 0 |
| 8.2 | 6.3 | 17.6 | 2.8 | 0 |
| 8.9 | 7.1 | 17.3 | 2.8 | 0 |
| 11 | 7.2 | 17.3 | 2.8 | 0 |
| 6 | 9 | 25 | 4.0 | 1 |
| 6.5 | 6 | 20 | 3.2 | 1 |
| 6 | 5.5 | 18 | 2.9 | 1 |
| 7 | 7 | 22 | 3.5 | 1 |
| 11.5 | 8.5 | 22 | 3.5 | 1 |
| 8.5 | 6 | 18 | 2.9 | 1 |
| 10.5 | 6.5 | 19 | 3.0 | 1 |
| 10.5 | 8.5 | 21 | 3.3 | 1 |
| 6.3 | 9.4 | 24 | 3.8 | 1 |
| 6.4 | 6.1 | 21 | 3.3 | 1 |
| 6.3 | 5.5 | 18.4 | 2.9 | 1 |
| 7 | 7.3 | 20 | 3.2 | 1 |
| 11.3 | 8.3 | 21.5 | 3.4 | 1 |
| 11.2 | 8.2 | 21 | 3.3 | 1 |
| 10 | 8 | 19 | 3.0 | 1 |
| 9 | 8 | 18 | 2.9 | 1 |
| 8.5 | 7 | 20 | 3.2 | 1 |
| 10 | 6 | 18 | 2.9 | 1 |
| 6 | 6.5 | 17.5 | 2.8 | 1 |
| 11.5 | 9 | 24 | 3.8 | 1 |
| 11.5 | 8.5 | 22 | 3.5 | 1 |
| 8.5 | 6 | 18 | 2.9 | 1 |
| 10.5 | 6.5 | 19 | 3.0 | 1 |
| 10.5 | 8.5 | 21 | 3.3 | 1 |
| 6.3 | 9.4 | 24 | 3.8 | 1 |
| 11 | 8 | 21 | 3.3 | 1 |
| 11.3 | 8 | 20 | 3.2 | 1 |
| 10.5 | 8 | 19.5 | 3.1 | 1 |
| 9.5 | 8 | 18 | 2.9 | 1 |
| 8.5 | 7 | 20 | 3.2 | 1 |
| 8.5 | 8 | 21 | 3.3 | 1 |
| 6.5 | 7 | 20 | 3.2 | 2 |
| 5.5 | 6.5 | 17 | 2.7 | 2 |
| 5.5 | 6.4 | 19.5 | 3.1 | 2 |
| 6 | 6.5 | 20.5 | 3.3 | 2 |
| 7 | 6.7 | 21.5 | 3.4 | 2 |
| 6.5 | 6.6 | 20.5 | 3.3 | 2 |
| 6.7 | 6.5 | 15.5 | 2.5 | 2 |
| 5.5 | 6.2 | 17 | 2.7 | 2 |
| 5.7 | 6.5 | 20 | 3.2 | 2 |
| 6 | 7 | 16.5 | 2.6 | 2 |
| 6.3 | 6.5 | 18 | 2.9 | 2 |
| 6 | 6.5 | 18.5 | 2.9 | 2 |
| 6.4 | 6.9 | 19 | 3.0 | 2 |
| 5.9 | 6.3 | 18.7 | 3.0 | 2 |
| 6.9 | 6.6 | 21.4 | 3.4 | 2 |
| 6.5 | 6.7 | 15.5 | 2.5 | 2 |
| 5.5 | 6.3 | 18 | 2.9 | 2 |
| 5.5 | 6.2 | 17 | 2.7 | 2 |
| 5.7 | 6.5 | 20 | 3.2 | 2 |
| 6 | 7 | 16.5 | 2.6 | 2 |
| 6.3 | 6.5 | 18 | 2.9 | 2 |
| 6 | 6.5 | 18.5 | 2.9 | 2 |
| 6 | 7 | 20 | 3.2 | 2 |
| 6.4 | 6 | 19 | 3.0 | 2 |
| 5.6 | 6.4 | 19.5 | 3.1 | 2 |
| 6.6 | 6 | 16 | 2.5 | 2 |
| 5.5 | 6.3 | 18 | 2.9 | 2 |
| 5.5 | 6.2 | 17 | 2.7 | 2 |
| 5.7 | 6.5 | 20 | 3.2 | 2 |
| 6 | 7 | 16.5 | 2.6 | 2 |

**Código:**

import numpy as np

import matplotlib.pyplot as plt

import pandas as pd

import mglearn

data= np.array([

[11,7,17,2.7 ],

[10,6.5,19,3.0 ],

[8,6,17.5,2.8 ],

[9,7,17.5,2.8 ],

[8.3,5,16,2.5 ],

[7.5,6,17.2,2.7 ],

[11.2,6.7,19.5,3.1 ],

[11.5,6,17.8,2.8 ],

[11.1,5.5,16.9,2.7 ],

[9,6,17.8,2.8 ],

[11.3,5.5,18,2.9 ],

[10.4,5.5,15.5,2.5 ],

[10.9,7.5,17.3,2.8 ],

[9,6,18.5,2.9 ],

[8.2,6.3,17.6,2.8 ],

[8.9,7.1,17.3,2.8 ],

[8.5,4.9,16.3,2.6 ],

[11,6.5,19,3.0 ],

[10.9,5.3,16.5,2.6 ],

[8.9,5.9,17.4,2.8 ],

[11.2,5.4,17,2.7 ],

[10.2,5.2,15.6,2.5 ],

[8.1,5.4,13.5,2.1 ],

[10.4,5.5,15.5,2.5 ],

[10.9,7.5,17.3,2.8 ],

[9,6,18.5,2.9 ],

[8.2,6.3,17.6,2.8 ],

[8.9,7.1,17.3,2.8 ],

[11,7.2,17.3,2.8 ],

[6,9,25,4.0 ],

[6.5,6,20,3.2 ],

[6,5.5,18,2.9 ],

[7,7,22,3.5 ],

[11.5,8.5,22,3.5 ],

[8.5,6,18,2.9 ],

[10.5,6.5,19,3.0 ],

[10.5,8.5,21,3.3 ],

[6.3,9.4,24,3.8 ],

[6.4,6.1,21,3.3 ],

[6.3,5.5,18.4,2.9 ],

[7,7.3,20,3.2 ],

[11.3,8.3,21.5,3.4 ],

[11.2,8.2,21,3.3 ],

[10,8,19,3.0 ],

[9,8,18,2.9 ],

[8.5,7,20,3.2 ],

[10,6,18,2.9 ],

[6,6.5,17.5,2.8 ],

[11.5,9,24,3.8 ],

[11.5,8.5,22,3.5 ],

[8.5,6,18,2.9 ],

[10.5,6.5,19,3.0 ],

[10.5,8.5,21,3.3 ],

[6.3,9.4,24,3.8 ],

[11,8,21,3.3 ],

[11.3,8,20,3.2 ],

[10.5,8,19.5,3.1 ],

[9.5,8,18,2.9 ],

[8.5,7,20,3.2 ],

[8.5,8,21,3.3 ],

[6.5,7,20,3.2 ],

[5.5,6.5,17,2.7 ],

[5.5,6.4,19.5,3.1 ],

[6,6.5,20.5,3.3 ],

[7,6.7,21.5,3.4 ],

[6.5,6.6,20.5,3.3 ],

[6.7,6.5,15.5,2.5 ],

[5.5,6.2,17,2.7 ],

[5.7,6.5,20,3.2 ],

[6,7,16.5,2.6 ],

[6.3,6.5,18,2.9 ],

[6,6.5,18.5,2.9 ],

[6.4,6.9,19,3.0 ],

[5.9,6.3,18.7,3.0 ],

[6.9,6.6,21.4,3.4 ],

[6.5,6.7,15.5,2.5 ],

[5.5,6.3,18,2.9 ],

[5.5,6.2,17,2.7 ],

[5.7,6.5,20,3.2 ],

[6,7,16.5,2.6 ],

[6.3,6.5,18,2.9 ],

[6,6.5,18.5,2.9 ],

[6,7,20,3.2 ],

[6.4,6,19,3.0 ],

[5.6,6.4,19.5,3.1 ],

[6.6,6,16,2.5 ],

[5.5,6.3,18,2.9 ],

[5.5,6.2,17,2.7 ],

[5.7,6.5,20,3.2 ],

[6,7,16.5,2.6 ]

]

)

target = np.array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,

1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,

2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2])

feature\_names = ['Alto', 'Ancho', 'Circunferencia', 'Radio']

target\_names = np.array(['Mante', 'Naranja', 'Manzana']);

dataset={

'data' : data,

'target' : target,

'target\_names' : target\_names,

'feature\_names' : feature\_names

}

print("Data shape: {}".format(data.shape))

print("Targets shape: {}".format(data.shape))

from sklearn.model\_selection import train\_test\_split

X\_train, X\_test, y\_train, y\_test = train\_test\_split(

dataset['data'], dataset['target'], random\_state=0)

print("X\_train shape: {}".format(X\_train.shape))

print("y\_train shape: {}".format(y\_train.shape))

print("X\_test shape: {}".format(X\_test.shape))

print("y\_test shape: {}".format(y\_test.shape))

dataframe = pd.DataFrame(X\_train, columns=dataset['feature\_names'])

grr = pd.plotting.scatter\_matrix(dataframe, c=y\_train, figsize=(15, 15), marker='o',

hist\_kwds={'bins': 20}, s=60, alpha=.8, cmap=mglearn.cm3)

from sklearn.neighbors import KNeighborsClassifier

knn = KNeighborsClassifier(n\_neighbors=1)

knn.fit(X\_train, y\_train)

#instancia desconocida

X\_new = np.array([[6.2,5.7,27.76,37.40 ]])

print("X\_new.shape: {}".format(X\_new.shape))

#Calculamos de que tipo será

prediction = knn.predict(X\_new)

print("Prediction: {}".format(prediction))

print("Predicted target name: {}".format( dataset['target\_names'][prediction]))

#Calculamos que tan precisa fue nuestra prediccion

print("Test set score: {:.2f}".format(knn.score(X\_test, y\_test)))

plt.show() #Mostramos los graficos

Explicación del Código:

Importamos las librerías necesarias:

import numpy as np

import matplotlib.pyplot as plt

import pandas as pd

import mglearn

Declaramos nuestros datos de entrada:

data= np.array([

[11, 7, 17,2.7 ],

[10,6.5,19,3.0 ],

[8,6,17.5,2.8 ],

[9,7,17.5,2.8 ],

[8.3,5,16,2.5 ],

[7.5,6,17.2,2.7 ],

[11.2,6.7,19.5,3.1 ],

[11.5,6,17.8,2.8 ],

[11.1,5.5,16.9,2.7 ],

[9,6,17.8,2.8 ],

[11.3,5.5,18,2.9 ],

[10.4,5.5,15.5,2.5 ],

[10.9,7.5,17.3,2.8 ],

[9,6,18.5,2.9 ],

[8.2,6.3,17.6,2.8 ],

[8.9,7.1,17.3,2.8 ],

[8.5,4.9,16.3,2.6 ],

[11,6.5,19,3.0 ],

[10.9,5.3,16.5,2.6 ],

[8.9,5.9,17.4,2.8 ],

[11.2,5.4,17,2.7 ],

[10.2,5.2,15.6,2.5 ],

[8.1,5.4,13.5,2.1 ],

[10.4,5.5,15.5,2.5 ],

[10.9,7.5,17.3,2.8 ],

[9,6,18.5,2.9 ],

[8.2,6.3,17.6,2.8 ],

[8.9,7.1,17.3,2.8 ],

[11,7.2,17.3,2.8 ],

[6,9,25,4.0 ],

[6.5,6,20,3.2 ],

[6,5.5,18,2.9 ],

[7,7,22,3.5 ],

[11.5,8.5,22,3.5 ],

[8.5,6,18,2.9 ],

[10.5,6.5,19,3.0 ],

[10.5,8.5,21,3.3 ],

[6.3,9.4,24,3.8 ],

[6.4,6.1,21,3.3 ],

[6.3,5.5,18.4,2.9 ],

[7,7.3,20,3.2 ],

[11.3,8.3,21.5,3.4 ],

[11.2,8.2,21,3.3 ],

[10,8,19,3.0 ],

[9,8,18,2.9 ],

[8.5,7,20,3.2 ],

[10,6,18,2.9 ],

[6,6.5,17.5,2.8 ],

[11.5,9,24,3.8 ],

[11.5,8.5,22,3.5 ],

[8.5,6,18,2.9 ],

[10.5,6.5,19,3.0 ],

[10.5,8.5,21,3.3 ],

[6.3,9.4,24,3.8 ],

[11,8,21,3.3 ],

[11.3,8,20,3.2 ],

[10.5,8,19.5,3.1 ],

[9.5,8,18,2.9 ],

[8.5,7,20,3.2 ],

[8.5,8,21,3.3 ],

[6.5,7,20,3.2 ],

[5.5,6.5,17,2.7 ],

[5.5,6.4,19.5,3.1 ],

[6,6.5,20.5,3.3 ],

[7,6.7,21.5,3.4 ],

[6.5,6.6,20.5,3.3 ],

[6.7,6.5,15.5,2.5 ],

[5.5,6.2,17,2.7 ],

[5.7,6.5,20,3.2 ],

[6,7,16.5,2.6 ],

[6.3,6.5,18,2.9 ],

[6,6.5,18.5,2.9 ],

[6.4,6.9,19,3.0 ],

[5.9,6.3,18.7,3.0 ],

[6.9,6.6,21.4,3.4 ],

[6.5,6.7,15.5,2.5 ],

[5.5,6.3,18,2.9 ],

[5.5,6.2,17,2.7 ],

[5.7,6.5,20,3.2 ],

[6,7,16.5,2.6 ],

[6.3,6.5,18,2.9 ],

[6,6.5,18.5,2.9 ],

[6,7,20,3.2 ],

[6.4,6,19,3.0 ],

[5.6,6.4,19.5,3.1 ],

[6.6,6,16,2.5 ],

[5.5,6.3,18,2.9 ],

[5.5,6.2,17,2.7 ],

[5.7,6.5,20,3.2 ],

[6,7,16.5,2.6 ]

]

)

Declaramos las clases de cada dato:

target = np.array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,

1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,

2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2])

Especificamos que significa cada valor:

// 0 1 2

target\_names = np.array(['Mante', 'Naranja', 'Manzana']);

Especificamos que atributos vamos a usar:

feature\_names = ['Alto', 'Ancho', 'Circunferencia', 'Radio']

Creamos nuestro diccionario de datos:

dataset={

'data' : data,

'target' : target,

'target\_names' : target\_names,

'feature\_names' : feature\_names

}

print("Data shape: {}".format(data.shape))

print("Targets shape: {}".format(data.shape))

Creamos nuestro set de entrenamiento y de pruebas:

from sklearn.model\_selection import train\_test\_split

X\_train, X\_test, y\_train, y\_test = train\_test\_split(

dataset['data'], dataset['target'], random\_state=0)

print("X\_train shape: {}".format(X\_train.shape))

print("y\_train shape: {}".format(y\_train.shape))

print("X\_test shape: {}".format(X\_test.shape))

print("y\_test shape: {}".format(y\_test.shape))

Si deseamos visualizar los datos entonces declaramos la creación de un grafico:

dataframe = pd.DataFrame(X\_train, columns=dataset['feature\_names'])

grr = pd.plotting.scatter\_matrix(dataframe, c=y\_train, figsize=(15, 15), marker='o',

hist\_kwds={'bins': 20}, s=60, alpha=.8, cmap=mglearn.cm3)

Creamos nuestro modelo de predicción

from sklearn.neighbors import KNeighborsClassifier

knn = KNeighborsClassifier(n\_neighbors=1)

knn.fit(X\_train, y\_train) // Y lo dotamos de datos de entrada

Declaramos nuestra instancia desconocida:

#instancia desconocida

X\_new = np.array([[6.2,5.7,27.76,37.40 ]])

print("X\_new.shape: {}".format(X\_new.shape))

Y realizamos la predicción:

#Calculamos de que tipo será

prediction = knn.predict(X\_new)

print("Prediction: {}".format(prediction))

print("Predicted target name: {}".format( dataset['target\_names'][prediction]))

Si deseamos calcular el nivel de confianza de nuestro algoritmo entonces:

#Calculamos que tan precisa fue nuestra prediccion

print("Test set score: {:.2f}".format(knn.score(X\_test, y\_test)))

plt.show() #Mostramos los graficos

Ejecución:



