TENEMOS UN SET DE DATOS DE VINOS, Y CADA FILA SE LE ASIGNA TARGET 0,1,2

SE TIENE 3 GRUPOS

SE TIENE UN SET DE TRAIN

Y SE EVALUA SI ESE APRENDIZAJE PUEDE PREDECIR A QUE GRUPO PERTENECE

SE LE COMPARA EL APRENDIJA VIENDO EL AJUSTE ENTRE ESOS DATOS CON EL TARGET QUE TIENE

SI DA UNA BUENA CERTEZA IMPLICA QUE TIENE UN MODELO BIEN ENTRENADO DE DATOS

PARA PODER HACER PREDICCIONES CON UN NUEVO GRUPO DE DATOS

INICIA CON UN SET QUE TIENE SEGURO EL GRUPO DE PROPIEDADES CON EL TARGET.

ES COMO DECIR:

* AQUI TENGO DATOS CERTEROS
* TE LOS DOY PARA QUE APRENDAS
* COMPARATE SI ESTAS SIENDO EXACTO
* DIME CUANTOS GRUPOS ENCONTRASTE

SI ES MUY EXACTO CON UN NUEVO GRUPO PREDECIRA MUCHO A QUE GRUPO PERTENECE

from sklearn.cluster import KMeans

import numpy as np

from sklearn import datasets

import pandas as pd

import matplotlib.pyplot as plt

from sklearn import metrics

DATOS = datasets.load\_wine()

# target 0,1,2

DataTarget = DATOS.target

col\_list = DATOS.feature\_names

DataToTrain = pd.DataFrame(DATOS.data, columns = col\_list)

print(col\_list)

K\_optimo = 0;

Mejor    = 0

#BUSCA EL MEJOR K ENTRE 2 Y 8

#EL MEJOR K ES CON MAS ACCURACY

#SE USA DataTrain Y SE COMPARA CON DataTarget para evaluar exactitude

for K in range(2,8):

    model = KMeans(n\_clusters= K, max\_iter=1000) #Entrenar modelo

    model.fit(DataToTrain)   #Entrena modelo

    y\_kmeans = model.predict(DataToTrain) #Creamos prediccion

    # Revisamos la precicion del modelo

    accuracy = metrics.adjusted\_rand\_score(DataTarget, y\_kmeans)

    print(K,' ',accuracy)

    if accuracy > Mejor:

        K\_optimo = K

        Mejor = accuracy

print('')

print(f'Max accuracy: {round(Mejor,3)}% \nUsando K: {K\_optimo} ')

Salida

['alcohol', 'malic\_acid', 'ash', 'alcalinity\_of\_ash', 'magnesium', 'total\_phenols', 'flavanoids', 'nonflavanoid\_phenols', 'proanthocyanins', 'color\_intensity', 'hue', 'od280/od315\_of\_diluted\_wines', 'proline']

2 0.3694075388574537

3 0.37111371823084754

4 0.303442531578347

5 0.31158801033176253

6 0.2909022768841822

7 0.22632105878612552

Max accuracy: 0.371%

Usando K: 3

K = K\_optimo

model = KMeans(n\_clusters= K, max\_iter=1000) #Crear modelo

model.fit(DataToTrain)   #Entrenar modelo

y\_means = model.predict(DataToTrain)  #Crear prediccion

#GRAFIQUE 2 FEATURES,

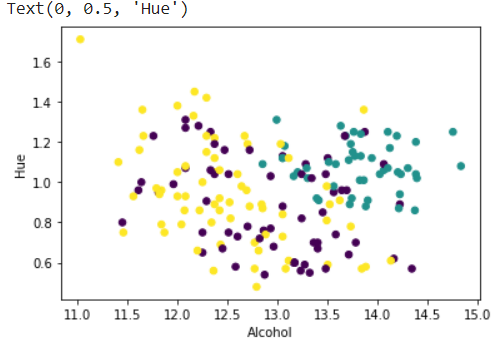
#A CADA PUNTO SE LE ASIGNA SU GRUPO SEGUN LA PREDICCION

plt.scatter(DataToTrain['alcohol'], DataToTrain['hue'],

c=y\_means, s=30)

plt.xlabel('Alcohol', fontsize = 10)

plt.ylabel('Hue', fontsize = 10)



INTERPRETACION:

y\_means tiene un lista de prediccion, de 0,1,2 y segun el valor le da color.

Esa prediccion dice a cada punto, pronostico que eres del grupo 0, 1 o 2, y se hace asignandole un color diferente.