

assignment-6

February 19, 2024

```
[12]: import pandas as pd
import numpy as np

df = pd.read_csv("/content/Iris (1).csv")
```

```
[13]: df.head()
```

```
[13]:
```

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
0	1	5.1	3.5	1.4	0.2	Iris-setosa
1	2	4.9	3.0	1.4	0.2	Iris-setosa
2	3	4.7	3.2	1.3	0.2	Iris-setosa
3	4	4.6	3.1	1.5	0.2	Iris-setosa
4	5	5.0	3.6	1.4	0.2	Iris-setosa

```
[14]: y = df['Species']
y.head()
```

```
[14]: 0    Iris-setosa
1    Iris-setosa
2    Iris-setosa
3    Iris-setosa
4    Iris-setosa
Name: Species, dtype: object
```

```
[15]: x = df.drop(['Species'], axis = 1)
x.head()
```

```
[15]:
```

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm
0	1	5.1	3.5	1.4	0.2
1	2	4.9	3.0	1.4	0.2
2	3	4.7	3.2	1.3	0.2
3	4	4.6	3.1	1.5	0.2
4	5	5.0	3.6	1.4	0.2

```
[16]: from sklearn.preprocessing import MinMaxScaler

scaler = MinMaxScaler()
```

```
x_scalar = scaler.fit_transform(x)
```

```
[17]: from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test = train_test_split(x_scalar,y,test_size=0.
↪25,random_state=125)
```

```
[18]: x_train
```

```
[18]: array([[0.53691275, 0.33333333, 0.16666667, 0.47457627, 0.41666667],
[0.05369128, 0.02777778, 0.375, 0.06779661, 0.04166667],
[0.06711409, 0.30555556, 0.70833333, 0.08474576, 0.04166667],
[0.91275168, 0.55555556, 0.58333333, 0.77966102, 0.95833333],
[0.20134228, 0.13888889, 0.45833333, 0.10169492, 0.04166667],
[0.69127517, 0.55555556, 0.375, 0.77966102, 0.70833333],
[0.69798658, 0.61111111, 0.41666667, 0.81355932, 0.875],
[0.14765101, 0.08333333, 0.66666667, 0., 0.04166667],
[0.82550336, 0.55555556, 0.29166667, 0.66101695, 0.70833333],
[0.32885906, 0.19444444, 0.54166667, 0.06779661, 0.04166667],
[0.79194631, 0.94444444, 0.25, 1., 0.91666667],
[0.67785235, 0.41666667, 0.29166667, 0.69491525, 0.75],
[0.95973154, 0.69444444, 0.5, 0.83050847, 0.91666667],
[0.63758389, 0.38888889, 0.41666667, 0.54237288, 0.45833333],
[0.46979866, 0.44444444, 0.5, 0.6440678, 0.70833333],
[0.48322148, 0.55555556, 0.20833333, 0.66101695, 0.58333333],
[0.36912752, 0.38888889, 0.33333333, 0.59322034, 0.5],
[0.8590604, 0.58333333, 0.33333333, 0.77966102, 0.83333333],
[0.99328859, 0.52777778, 0.58333333, 0.74576271, 0.91666667],
[0.55033557, 0.41666667, 0.29166667, 0.49152542, 0.45833333],
[0.93959732, 0.66666667, 0.45833333, 0.77966102, 0.95833333],
[0.04026846, 0.08333333, 0.58333333, 0.06779661, 0.08333333],
[0.85234899, 0.5, 0.41666667, 0.66101695, 0.70833333],
[0.01342282, 0.11111111, 0.5, 0.05084746, 0.04166667],
[0.59060403, 0.36111111, 0.41666667, 0.52542373, 0.5],
[0.55704698, 0.47222222, 0.29166667, 0.69491525, 0.625],
[0.1409396, 0.22222222, 0.70833333, 0.08474576, 0.125],
[0.46308725, 0.36111111, 0.20833333, 0.49152542, 0.41666667],
[0.6442953, 0.38888889, 0.375, 0.54237288, 0.5],
[0.08053691, 0.13888889, 0.41666667, 0.06779661, 0.],
[0.86577181, 0.80555556, 0.41666667, 0.81355932, 0.625],
[0.41610738, 0.47222222, 0.08333333, 0.50847458, 0.375],
[0.63087248, 0.36111111, 0.29166667, 0.54237288, 0.5],
[0.66442953, 0.38888889, 0.33333333, 0.52542373, 0.5],
[0.70469799, 0.91666667, 0.41666667, 0.94915254, 0.83333333],
[0.18120805, 0.25, 0.625, 0.08474576, 0.04166667],
[0.62416107, 0.19444444, 0.125, 0.38983051, 0.375],
[0.80536913, 0.72222222, 0.5, 0.79661017, 0.91666667],
[0.7114094, 0.16666667, 0.20833333, 0.59322034, 0.66666667],
```

[0.79865772, 0.47222222, 0.08333333, 0.6779661 , 0.58333333],
 [0.65100671, 0.52777778, 0.375 , 0.55932203, 0.5],
 [0.56375839, 0.30555556, 0.41666667, 0.59322034, 0.58333333],
 [0.48993289, 0.5 , 0.33333333, 0.62711864, 0.45833333],
 [0.28187919, 0.02777778, 0.5 , 0.05084746, 0.04166667],
 [0.53020134, 0.38888889, 0.25 , 0.42372881, 0.375],
 [0.32214765, 0.27777778, 0.70833333, 0.08474576, 0.04166667],
 [0.7852349 , 0.94444444, 0.75 , 0.96610169, 0.875],
 [0.94630872, 0.72222222, 0.45833333, 0.69491525, 0.91666667],
 [0.75167785, 0.69444444, 0.41666667, 0.76271186, 0.83333333],
 [0.16778523, 0.19444444, 0.41666667, 0.10169492, 0.04166667],
 [0.51677852, 0.66666667, 0.41666667, 0.6779661 , 0.66666667],
 [0.20805369, 0.30555556, 0.58333333, 0.08474576, 0.125],
 [0.00671141, 0.16666667, 0.41666667, 0.06779661, 0.04166667],
 [0.96644295, 0.66666667, 0.54166667, 0.79661017, 1.],
 [0.12080537, 0.38888889, 0.75 , 0.11864407, 0.08333333],
 [0.17449664, 0.19444444, 0.58333333, 0.10169492, 0.125],
 [0.10738255, 0.30555556, 0.79166667, 0.05084746, 0.125],
 [0.68456376, 0.77777778, 0.41666667, 0.83050847, 0.83333333],
 [0.16107383, 0.13888889, 0.58333333, 0.15254237, 0.04166667],
 [0.81208054, 0.36111111, 0.33333333, 0.66101695, 0.79166667],
 [0.03355705, 0.30555556, 0.79166667, 0.11864407, 0.125],
 [0.91946309, 0.58333333, 0.45833333, 0.76271186, 0.70833333],
 [0.22818792, 0.16666667, 0.45833333, 0.08474576, 0.],
 [0.97315436, 0.66666667, 0.41666667, 0.71186441, 0.91666667],
 [0.61073826, 0.5 , 0.41666667, 0.61016949, 0.54166667],
 [0.81879195, 0.94444444, 0.33333333, 0.96610169, 0.79166667],
 [0.13422819, 0.30555556, 0.58333333, 0.11864407, 0.04166667],
 [0.38926174, 0.63888889, 0.375 , 0.61016949, 0.5],
 [0.43624161, 0.66666667, 0.45833333, 0.57627119, 0.54166667],
 [0.74496644, 0.58333333, 0.29166667, 0.72881356, 0.75],
 [0.15436242, 0.22222222, 0.54166667, 0.11864407, 0.16666667],
 [0.90604027, 0.94444444, 0.41666667, 0.86440678, 0.91666667],
 [0.52348993, 0.47222222, 0.375 , 0.59322034, 0.58333333],
 [0.95302013, 0.41666667, 0.29166667, 0.69491525, 0.75],
 [0.87248322, 0.86111111, 0.33333333, 0.86440678, 0.75],
 [0.89261745, 0.55555556, 0.33333333, 0.69491525, 0.58333333],
 [0.5033557 , 0.63888889, 0.41666667, 0.57627119, 0.54166667],
 [0.24161074, 0.33333333, 0.625 , 0.05084746, 0.04166667],
 [0.11409396, 0.22222222, 0.625 , 0.06779661, 0.08333333],
 [0.23489933, 0.19444444, 0.5 , 0.03389831, 0.04166667],
 [0.25503356, 0.02777778, 0.41666667, 0.05084746, 0.04166667],
 [0.83221477, 0.66666667, 0.54166667, 0.79661017, 0.83333333],
 [0.98657718, 0.61111111, 0.41666667, 0.71186441, 0.79166667],
 [0.71812081, 0.83333333, 0.375 , 0.89830508, 0.70833333],
 [0.45637584, 0.52777778, 0.08333333, 0.59322034, 0.58333333],
 [0.02684564, 0.19444444, 0.66666667, 0.06779661, 0.04166667],

```
[0.54362416, 0.33333333, 0.16666667, 0.45762712, 0.375      ],
[0.89932886, 0.5          , 0.25          , 0.77966102, 0.54166667],
[0.27516779, 0.05555556, 0.125          , 0.05084746, 0.08333333],
[0.18791946, 0.25          , 0.58333333, 0.06779661, 0.04166667],
[0.04697987, 0.19444444, 0.58333333, 0.08474576, 0.04166667],
[0.29530201, 0.22222222, 0.75          , 0.15254237, 0.125      ],
[0.31543624, 0.08333333, 0.5          , 0.06779661, 0.04166667],
[0.51006711, 0.69444444, 0.33333333, 0.6440678 , 0.54166667],
[0.87919463, 1.          , 0.75          , 0.91525424, 0.79166667],
[0.84563758, 0.52777778, 0.33333333, 0.6440678 , 0.70833333],
[0.06040268, 0.16666667, 0.45833333, 0.08474576, 0.          ],
[0.61744966, 0.41666667, 0.25          , 0.50847458, 0.45833333],
[0.44295302, 0.36111111, 0.41666667, 0.59322034, 0.58333333],
[1.          , 0.44444444, 0.41666667, 0.69491525, 0.70833333],
[0.59731544, 0.33333333, 0.20833333, 0.50847458, 0.5          ],
[0.72483221, 0.66666667, 0.20833333, 0.81355932, 0.70833333],
[0.33557047, 0.75          , 0.5          , 0.62711864, 0.54166667],
[0.          , 0.22222222, 0.625          , 0.06779661, 0.04166667],
[0.57718121, 0.66666667, 0.45833333, 0.62711864, 0.58333333],
[0.24832215, 0.16666667, 0.45833333, 0.08474576, 0.          ],
[0.4295302 , 0.36111111, 0.375          , 0.44067797, 0.5          ],
[0.39597315, 0.25          , 0.29166667, 0.49152542, 0.54166667],
[0.2885906 , 0.19444444, 0.625          , 0.10169492, 0.20833333],
[0.73825503, 0.61111111, 0.5          , 0.69491525, 0.79166667],
[0.44966443, 0.41666667, 0.29166667, 0.52542373, 0.375      ],
[0.58389262, 0.55555556, 0.125          , 0.57627119, 0.5          ]])
```

```
[19]: from sklearn.naive_bayes import GaussianNB
      gnb = GaussianNB()
      gnb.fit(x_train, y_train)
```

```
[19]: GaussianNB()
```

```
[20]: y_pred = gnb.predict(x_test)
```

```
[21]: from sklearn.metrics import
      ↪confusion_matrix, accuracy_score, precision_score, recall_score
      print(accuracy_score(y_test, y_pred))
```

```
1.0
```

```
[22]: CM = confusion_matrix(y_test, y_pred)
      CM
```

```
[22]: array([[13,  0,  0],
            [ 0, 14,  0],
            [ 0,  0, 11]])
```

```
[23]: print(precision_score(y_test, y_pred, average='micro'))
```

1.0

```
[24]: print(recall_score(y_test, y_pred, average= 'micro'))
```

1.0

```
[25]: from sklearn.model_selection import cross_val_score  
accuracy_score = cross_val_score(gnb, x_scalar,y, cv=5)  
print(accuracy_score)
```

[0.96666667 1. 1. 1. 1.]

```
[25]:
```