

pratical-4

February 18, 2024

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

```
[42]: df = pd.read_csv('Iris.csv')
```

```
[43]: df.head(5)
```

```
[43]:
```

	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

```
[44]: df.drop(['Id'],axis=1,inplace=True)
```

```
[45]: df.iloc[:,4:]
```

```
[45]:
```

	Species
0	Iris-setosa
1	Iris-setosa
2	Iris-setosa
3	Iris-setosa
4	Iris-setosa
...	...
145	Iris-virginica
146	Iris-virginica
147	Iris-virginica
148	Iris-virginica
149	Iris-virginica

[150 rows x 1 columns]

```
[46]: from sklearn.preprocessing import MinMaxScaler
ms = MinMaxScaler()
ms.fit(df.drop('Species',axis=1))
```

```
[46]: MinMaxScaler()
```

```
[47]: df_scaled = pd.DataFrame(ms.transform(df.drop('Species',axis=1)),columns=df.  
    ↪iloc[:, :-1].columns)
```

```
[48]: df
```

```
[48]:
```

	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
0	5.1	3.5	1.4	0.2	Iris-setosa
1	4.9	3.0	1.4	0.2	Iris-setosa
2	4.7	3.2	1.3	0.2	Iris-setosa
3	4.6	3.1	1.5	0.2	Iris-setosa
4	5.0	3.6	1.4	0.2	Iris-setosa
..
145	6.7	3.0	5.2	2.3	Iris-virginica
146	6.3	2.5	5.0	1.9	Iris-virginica
147	6.5	3.0	5.2	2.0	Iris-virginica
148	6.2	3.4	5.4	2.3	Iris-virginica
149	5.9	3.0	5.1	1.8	Iris-virginica

[150 rows x 5 columns]

```
[50]: from sklearn.model_selection import train_test_split  
X_train, X_test, y_train, y_test = train_test_split(df_scaled, df.iloc[:,4:],  
    ↪test_size=0.20, random_state=20)
```

```
[53]: X_train.shape
```

```
[53]: (120, 4)
```

```
[54]: X_test.shape
```

```
[54]: (30, 4)
```

```
[55]: from sklearn.naive_bayes import GaussianNB  
gnb = GaussianNB()  
y_pred = gnb.fit(X_train, y_train).predict(X_test)
```

```
/home/student/anaconda3/lib/python3.9/site-  
packages/sklearn/utils/validation.py:993: DataConversionWarning: A column-vector  
y was passed when a 1d array was expected. Please change the shape of y to  
(n_samples, ), for example using ravel().  
  y = column_or_1d(y, warn=True)
```

```
[57]: y_pred
```

```
[57]: array(['Iris-setosa', 'Iris-versicolor', 'Iris-versicolor',
        'Iris-virginica', 'Iris-versicolor', 'Iris-versicolor',
        'Iris-virginica', 'Iris-setosa', 'Iris-virginica', 'Iris-setosa',
        'Iris-virginica', 'Iris-versicolor', 'Iris-versicolor',
        'Iris-setosa', 'Iris-setosa', 'Iris-virginica', 'Iris-setosa',
        'Iris-versicolor', 'Iris-virginica', 'Iris-versicolor',
        'Iris-versicolor', 'Iris-virginica', 'Iris-virginica',
        'Iris-setosa', 'Iris-versicolor', 'Iris-versicolor',
        'Iris-versicolor', 'Iris-setosa', 'Iris-virginica',
        'Iris-versicolor'], dtype='<U15')
```

```
[62]: from sklearn.metrics import confusion_matrix
conf_matrix = confusion_matrix(y_test, y_pred)

conf_matrix
```

```
[62]: array([[ 8,  0,  0],
        [ 0, 11,  0],
        [ 0,  2,  9]])
```

```
[80]: from sklearn.metrics import accuracy_score, recall_score, precision_score

print('Accuracy : {}'.format(accuracy_score(y_test, y_pred).round(2)))
```

Accuracy : 0.93

```
[82]: print('precision : {}'.format(precision_score(y_test, y_
↪ y_pred, average='weighted').round(2)))
```

precision : 0.94

```
[83]: print('Recall : {}'.format(recall_score(y_test, y_pred, average='weighted').
↪ round(2)))
```

Recall : 0.93

```
[ ]:
```