practice-assignment-6

May 4, 2025

[1]:
```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
```

[37]:
```
vm=pd.read_csv("iris.csv")
vm
```

[37]:
```
      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
..   …            …           …            …           …
145  146            6.7           3.0            5.2           2.3
146  147            6.3           2.5            5.0           1.9
147  148            6.5           3.0            5.2           2.0
148  149            6.2           3.4            5.4           2.3
149  150            5.9           3.0            5.1           1.8

            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 6 columns]
```

[7]:

```
print(vm.head())
print(vm.tail())
print(vm.info())
print(vm.describe())
print(vm.shape)
print(vm.size)
```

```
   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
      Id  SepalLengthCm  SepalWidthCm  PetalLengthCm  PetalWidthCm  \
145  146            6.7           3.0            5.2           2.3
146  147            6.3           2.5            5.0           1.9
147  148            6.5           3.0            5.2           2.0
148  149            6.2           3.4            5.4           2.3
149  150            5.9           3.0            5.1           1.8

            Species
145  Iris-virginica
146  Iris-virginica
147  Iris-virginica
148  Iris-virginica
149  Iris-virginica
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 150 entries, 0 to 149
Data columns (total 6 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   Id             150 non-null    int64
 1   SepalLengthCm  150 non-null    float64
 2   SepalWidthCm   150 non-null    float64
 3   PetalLengthCm  150 non-null    float64
 4   PetalWidthCm   150 non-null    float64
 5   Species        150 non-null    object
dtypes: float64(4), int64(1), object(1)
memory usage: 7.2+ KB
None
               Id  SepalLengthCm  SepalWidthCm  PetalLengthCm  PetalWidthCm
count  150.000000     150.000000    150.000000     150.000000    150.000000
mean    75.500000       5.843333      3.054000       3.758667      1.198667
std     43.445368       0.828066      0.433594       1.764420      0.763161
min      1.000000       4.300000      2.000000       1.000000      0.100000
25%     38.250000       5.100000      2.800000       1.600000      0.300000
50%     75.500000       5.800000      3.000000       4.350000      1.300000
```

```
75%    112.750000       6.400000      3.300000       5.100000      1.800000
max    150.000000       7.900000      4.400000       6.900000      2.500000
(150, 6)
900
```

[9]:

```
print(vm.isnull())
print(vm.isnull().sum())
```

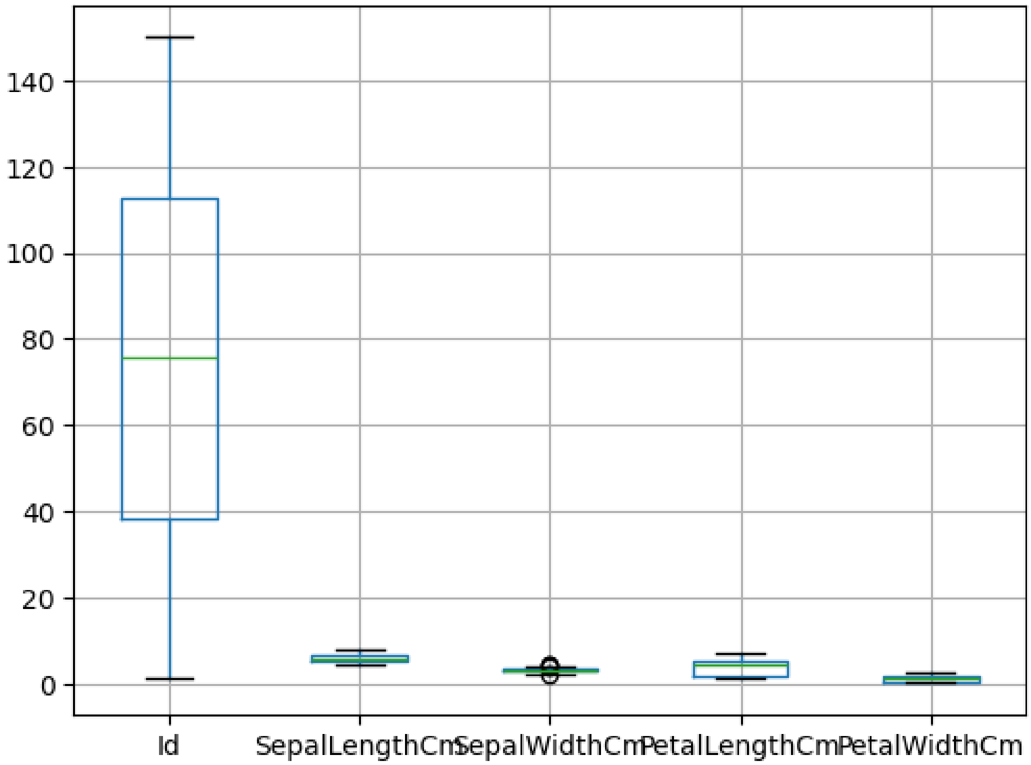
```
        Id  SepalLengthCm  SepalWidthCm  PetalLengthCm  PetalWidthCm  Species
0    False          False         False          False         False    False
1    False          False         False          False         False    False
2    False          False         False          False         False    False
3    False          False         False          False         False    False
4    False          False         False          False         False    False
..     …             …           …            …           …      …
145  False          False         False          False         False    False
146  False          False         False          False         False    False
147  False          False         False          False         False    False
148  False          False         False          False         False    False
149  False          False         False          False         False    False

[150 rows x 6 columns]
Id               0
SepalLengthCm    0
SepalWidthCm     0
PetalLengthCm    0
PetalWidthCm     0
Species          0
dtype: int64
```

[11]:

```
vm.boxplot()
```

[11]:

```
<Axes: >
```



[15]:
```
Q1 = vm['SepalWidthCm'].quantile(0.25)
Q3 = vm['SepalWidthCm'].quantile(0.75)
IQR = Q3 - Q1
Lower_limit = Q1 - 1.5 * IQR
Upper_limit = Q3 + 1.5 * IQR
print(f'Q1 = {Q1}, Q3 = {Q3}, IQR = {IQR}, Lower_limit =␣
 ↪{Lower_limit},␣Upper_limit = {Upper_limit}')
```

```
Q1 = 2.8, Q3 = 3.3, IQR = 0.5, Lower_limit = 2.05,␣Upper_limit = 4.05
```

[17]:
```
outliers=[]
for i in vm.SepalWidthCm:
    if i<Lower_limit or i>Upper_limit:
        outliers.append(i)
outliers
```

[17]:
```
[4.4, 4.1, 4.2, 2.0]
```

[31]:
```
out_ind=vm[(vm.SepalWidthCm<Lower_limit)|(vm.SepalWidthCm>Upper_limit)].index
df1=vm.drop(out_ind)
```

[43]:
```
df1
```

[43]:

```
      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
..   …              …            …             …            …
145  146            6.7           3.0            5.2           2.3
146  147            6.3           2.5            5.0           1.9
147  148            6.5           3.0            5.2           2.0
148  149            6.2           3.4            5.4           2.3
149  150            5.9           3.0            5.1           1.8

            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

[146 rows x 6 columns]
```

[47]:

```
df1.boxplot()
```

[47]:

```
<Axes: >
```

[53]:

```
outliers_sw=[]
for i in df1.SepalWidthCm:
    if i<Lower_limit or i>Upper_limit:
        outliers_sw.append(i)
outliers_sw
```

[53]:

```
[]
```

[59]:

```
#divide the dataset into independent(X) and dependent variables (Y)
X = df1.drop(['Species'], axis = 1)
Y = df1['Species']
print(X)
print(Y)
```

```
      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
..   …            …           …             …           …
145  146            6.7           3.0            5.2           2.3
146  147            6.3           2.5            5.0           1.9
147  148            6.5           3.0            5.2           2.0
148  149            6.2           3.4            5.4           2.3
149  150            5.9           3.0            5.1           1.8

[146 rows x 5 columns]
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
Name: Species, Length: 146, dtype: object
```

[61]:

```
#split the data into training and testing sets
from sklearn. model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split(X, Y, test_size = 0.2,␣
 ↪random_state = 0)
```

[63]:

```
from sklearn.naive_bayes import GaussianNB
classifier = GaussianNB()
classifier.fit(x_train, y_train)
```

[63]:

```
GaussianNB()
```

[65]:

```
ypred = classifier.predict(x_test)
```

[67]:

```
from sklearn.metrics import accuracy_score, precision_score,␣
 ↪recall_score,confusion_matrix,classification_report
```

[71]:

```
accuracy = accuracy_score(y_test,ypred)
print('Accuracy',accuracy)
```

```
Accuracy 1.0
```

[73]:

```
#compute the confusion matrix
cm=confusion_matrix(y_test, ypred)
print("Confusion Matrix:-",cm)
```

```
Confusion Matrix:- [[11  0  0]
 [ 0 10  0]
 [ 0  0  9]]
```

[75]:

```
pscore=precision_score(y_test, ypred,average='micro')
print("Precision Score:-",pscore)
```

```
Precision Score:- 1.0
```

[77]:

```
recalls=recall_score(y_test, ypred,average='micro')
print("Recall Score:-",recalls)
```

```
Recall Score:- 1.0
```

[79]:

```
error_rate=1-accuracy_score(y_test, ypred)
print("Error Rate:-",error_rate)
```

```
Error Rate:- 0.0
```

[81]:

```
print("Classification Report",classification_report(y_test, ypred))
```

```
Classification Report                  precision    recall  f1-score   support

    Iris-setosa       1.00      1.00      1.00        11
Iris-versicolor       1.00      1.00      1.00        10
 Iris-virginica       1.00      1.00      1.00         9

       accuracy                           1.00        30
      macro avg       1.00      1.00      1.00        30
   weighted avg       1.00      1.00      1.00        30
```

[ ]:

```
```