smartinternz-assignment-3

July 25, 2024

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

```
[43]: data=pd.read_csv('/content/Iris (1).csv')
```

```
[44]: data.head()
```

```
[44]:    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
```

```
[45]: data.shape
```

```
[45]: (150, 6)
```

```
[46]: data.info()
```

```
<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
```

```
[47]: data.tail()
```

```
[47]:       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
```

1 It is a Classification Model data

```
[48]: data.isnull().sum()
```

```
[48]: Id               0
      SepalLengthCm    0
      SepalWidthCm     0
      PetalLengthCm    0
      PetalWidthCm     0
      Species          0
      dtype: int64
```

```
[ ]: data1 = data.drop(columns=['Id'],axis=1)
```

```
[49]: data1
```

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

      [150 rows x 5 columns]
```

```
[50]: from sklearn.preprocessing import LabelEncoder
```

[51]:
```
le=LabelEncoder()
```

[52]:
```
data1.Species=le.fit_transform(data1.Species)
```

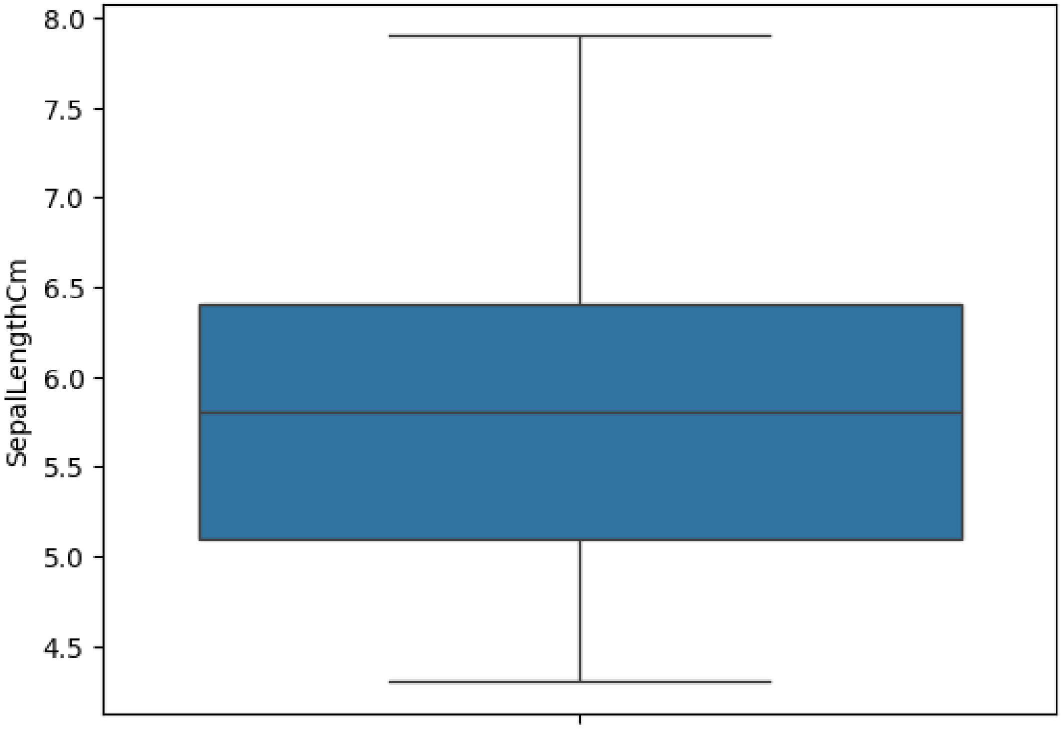
[53]:
```
data1.head()
```

[53]:
```
   SepalLengthCm  SepalWidthCm  PetalLengthCm  PetalWidthCm  Species
0            5.1           3.5            1.4           0.2        0
1            4.9           3.0            1.4           0.2        0
2            4.7           3.2            1.3           0.2        0
3            4.6           3.1            1.5           0.2        0
4            5.0           3.6            1.4           0.2        0
```

[54]:
```
import seaborn as sns
```

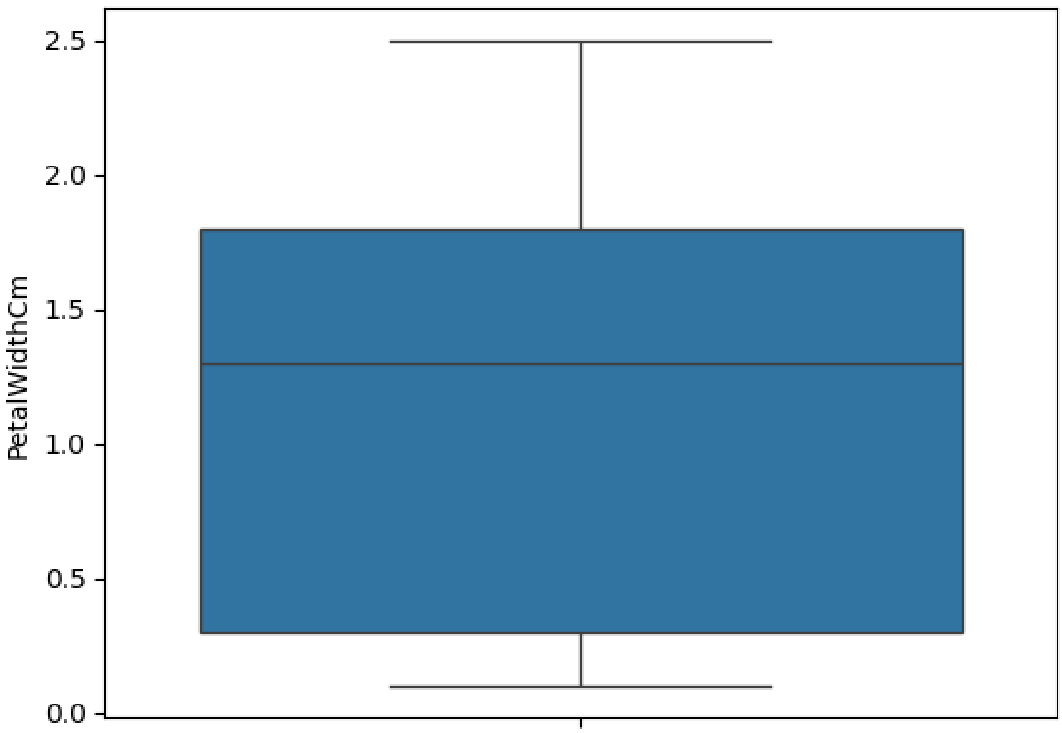
[55]:
```
sns.boxplot(data1.SepalLengthCm)
```

[55]:
```
<Axes: ylabel='SepalLengthCm'>
```



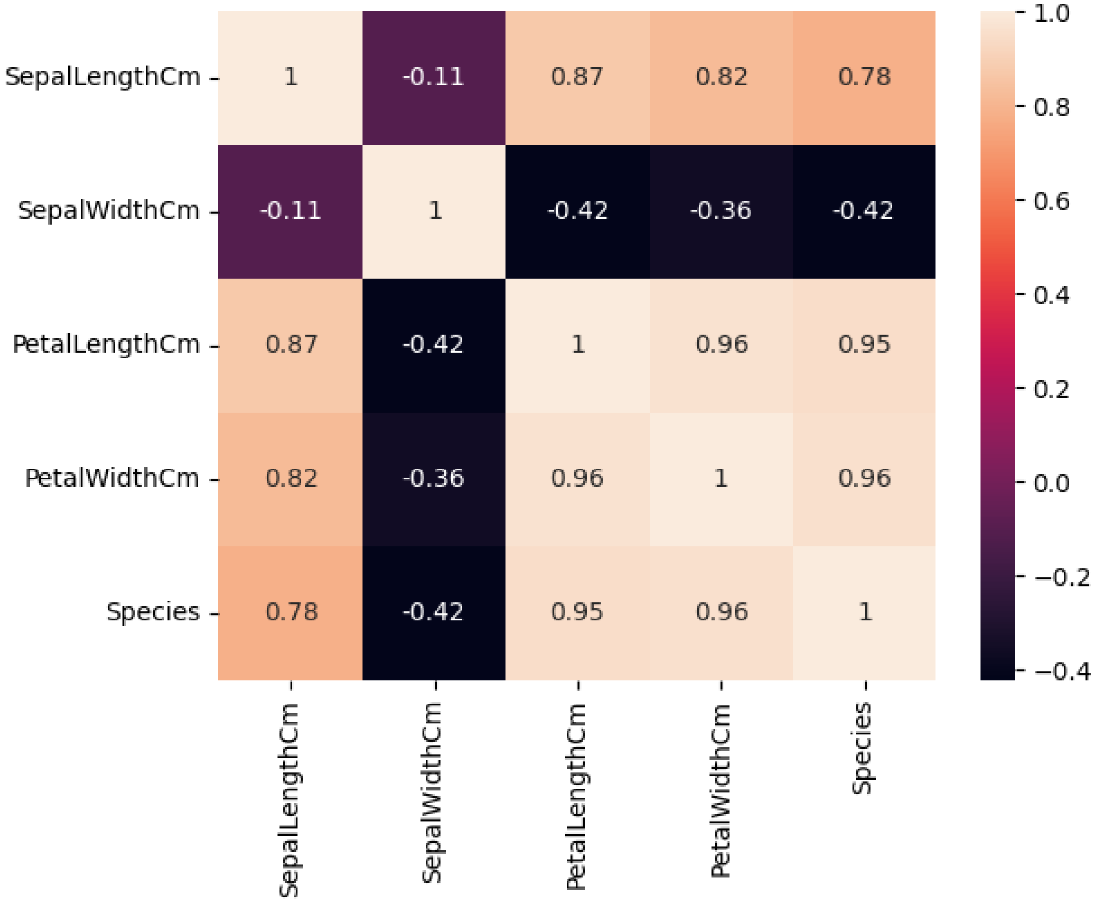
[56]:
```
sns.boxplot(data1.PetalWidthCm)
```

[56]:
```
<Axes: ylabel='PetalWidthCm'>
```



```
[57]: sns.heatmap(data1.corr(),annot=True)
```

```
[57]: <Axes: >
```



```
[58]: x=data1.drop('Species',axis=1)
      x
```

```
[58]:      SepalLengthCm  SepalWidthCm  PetalLengthCm  PetalWidthCm
      0              5.1           3.5            1.4           0.2
      1              4.9           3.0            1.4           0.2
      2              4.7           3.2            1.3           0.2
      3              4.6           3.1            1.5           0.2
      4              5.0           3.6            1.4           0.2
      ..             …             …              …             …
      145            6.7           3.0            5.2           2.3
      146            6.3           2.5            5.0           1.9
      147            6.5           3.0            5.2           2.0
      148            6.2           3.4            5.4           2.3
      149            5.9           3.0            5.1           1.8

      [150 rows x 4 columns]
```

[59]:

```
y=data1['Species']
y
```

[59]:

```
0      0
1      0
2      0
3      0
4      0
      ..
145    2
146    2
147    2
148    2
149    2
Name: Species, Length: 150, dtype: int64
```

[60]:

```
from sklearn.preprocessing import MinMaxScaler
Scale=MinMaxScaler()
```

[61]:

```
Scaled_x=pd.DataFrame(Scale.fit_transform(x),columns=x.columns)
```

[62]:

```
Scaled_x
```

[62]:

```
     SepalLengthCm  SepalWidthCm  PetalLengthCm  PetalWidthCm
0         0.222222      0.625000       0.067797      0.041667
1         0.166667      0.416667       0.067797      0.041667
2         0.111111      0.500000       0.050847      0.041667
3         0.083333      0.458333       0.084746      0.041667
4         0.194444      0.666667       0.067797      0.041667
..             …             …             …             …
145       0.666667      0.416667       0.711864      0.916667
146       0.555556      0.208333       0.677966      0.750000
147       0.611111      0.416667       0.711864      0.791667
148       0.527778      0.583333       0.745763      0.916667
149       0.444444      0.416667       0.694915      0.708333

[150 rows x 4 columns]
```

[63]:

```
from sklearn.model_selection import train_test_split
```

[66]:

```
x_train, x_test, y_train, y_test = train_test_split(Scaled_x, y, test_size=0.2,␣
 ↪random_state=42)
```

[67]:

```
x_train.shape
```

[67]:

```
(120, 4)
```

```
[68]: from sklearn.linear_model import LogisticRegression
```

```
[69]: model=LogisticRegression()
```

```
[71]: model.fit(x_train,y_train)
```

```
[71]: LogisticRegression()
```

```
[72]: pred=model.predict(x_test)
      pred
```

```
[72]: array([1, 0, 2, 1, 1, 0, 1, 2, 1, 1, 2, 0, 0, 0, 0, 2, 2, 1, 1, 2, 0, 2,
             0, 2, 2, 2, 2, 2, 0, 0])
```

```
[73]: y_test
```

```
[73]: 73     1
      18     0
      118    2
      78     1
      76     1
      31     0
      64     1
      141    2
      68     1
      82     1
      110    2
      12     0
      36     0
      9      0
      19     0
      56     1
      104    2
      69     1
      55     1
      132    2
      29     0
      127    2
      26     0
      128    2
      131    2
      145    2
      108    2
      143    2
      45     0
      30     0
      Name: Species, dtype: int64
```

```
[74]: data1.head()
```

```
[74]:    SepalLengthCm  SepalWidthCm  PetalLengthCm  PetalWidthCm  Species
      0            5.1           3.5            1.4           0.2        0
      1            4.9           3.0            1.4           0.2        0
      2            4.7           3.2            1.3           0.2        0
      3            4.6           3.1            1.5           0.2        0
      4            5.0           3.6            1.4           0.2        0
```

```
[75]: model.predict([[5.1,3.5,1.4,0.2]])
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/base.py:439: UserWarning: X does
not have valid feature names, but LogisticRegression was fitted with feature
names
  warnings.warn(
```

```
[75]: array([2])
```

```
[76]: from sklearn.metrics import accuracy_score
```

```
[77]: accuracy_score(y_test,pred)
```

```
[77]: 0.9666666666666667
```