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

```
D:\NEWANACO\lib\site-packages\statsmodels\tools\_testing.py:19: FutureWarning: pandas.util.testing is deprecated. Use the functions in the public API at pandas.testing instead.
  import pandas.util.testing as tm
```

In [2]:
```
data=pd.read_csv(r'C:\Users\karth\Desktop\iris\iris.csv')
```

In [3]:
```
data.head()
```

Out[3]:

| | 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 |

In [4]:
```
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
```

In [5]:
```
data['Species'].value_counts()
```

Out[5]:
```
Iris-setosa        50
Iris-virginica     50
Iris-versicolor    50
Name: Species, dtype: int64
```

In [6]:
```
data.describe()
```

Out[6]:

| | 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 |

In [7]:
```
data.skew()
```

Out[7]:
```
Id               0.000000
SepalLengthCm    0.314911
SepalWidthCm     0.334053
PetalLengthCm   -0.274464
PetalWidthCm    -0.104997
dtype: float64
```

In [8]:
```
data.kurtosis()
```

Out[8]:
```
Id              -1.200000
SepalLengthCm   -0.552064
SepalWidthCm     0.290781
PetalLengthCm   -1.401921
PetalWidthCm    -1.339754
dtype: float64
```

In [9]:
```
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
```

In [10]:
```
lbl=LinearRegression()
```

In [11]:
```
target=data['Species']
```

In [12]:
```
data_train=data.drop(['Species','Id'],axis=1)
```

In [13]:
```
data_train.head()
```

Out[13]:

| | 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 |

In [14]:
```
x_train,x_test,y_train,y_test=train_test_split(data_train,target,test_size=0.33,random_state=40)
```

In [15]:
```
x_train.shape
```

Out[15]:
```
(100, 4)
```

In [16]:
```
y_train.shape
```

Out[16]:
```
(100,)
```

In [17]:
```
x_train.head()
```

Out[17]:

| | SepalLengthCm | SepalWidthCm | PetalLengthCm | PetalWidthCm |
|---|---|---|---|---|
| **100** | 6.3 | 3.3 | 6.0 | 2.5 |
| **54** | 6.5 | 2.8 | 4.6 | 1.5 |
| **67** | 5.8 | 2.7 | 4.1 | 1.0 |
| **97** | 6.2 | 2.9 | 4.3 | 1.3 |
| **24** | 4.8 | 3.4 | 1.9 | 0.2 |

In [18]:
```
y_train.head()
```

Out[18]:
```
100     Iris-virginica
54     Iris-versicolor
67     Iris-versicolor
97     Iris-versicolor
24         Iris-setosa
Name: Species, dtype: object
```

In [19]:
```
convert_label = {"Iris-versicolor": 0, "Iris-setosa": 1, "Iris-virginica": 2}
target = target.map(convert_label)
```

In [20]:
```
target.head()
```

Out[20]:
```
0    1
1    1
2    1
3    1
4    1
Name: Species, dtype: int64
```

In [21]:
```
x_train,x_test,y_train,y_test=train_test_split(data_train,target,test_size=0.33,random_state=40)
```

In [22]:
```
y_train.head()
```

Out[22]:
```
100    2
54     0
67     0
97     0
24     1
Name: Species, dtype: int64
```

In [23]:
```
lbl.fit(x_train,y_train)
```

Out[23]:
```
LinearRegression(copy_X=True, fit_intercept=True, n_jobs=None, normalize=False)
```

In [24]:
```
lbl.score(x_train,y_train)
```

Out[24]:
```
0.2283658753146547
```

In [25]:
```
lbl.score(x_test,y_test)
```

Out[25]:
```
0.3719269819730897
```

In [26]:
```
from sklearn.neighbors import KNeighborsClassifier
```

In [27]:
```
knn=KNeighborsClassifier(n_neighbors=3)
```

In [28]:
```
knn.fit(x_train,y_train)
```

Out[28]:
```
KNeighborsClassifier(algorithm='auto', leaf_size=30, metric='minkowski',
                     metric_params=None, n_jobs=None, n_neighbors=3, p=2,
                     weights='uniform')
```

In [29]:
```
knn.score(x_train,y_train)
```

Out[29]:
```
0.96
```

In [30]:
```
knn.score(x_test,y_test)
```

Out[30]:
```
0.96
```

In [31]:
```
pred=knn.predict(x_test)
```

In [32]:
```
from sklearn.metrics import classification_report,confusion_matrix
from sklearn.model_selection import cross_val_score
```

In [33]:
```
print(classification_report(y_test,pred))
```

```
              precision    recall  f1-score   support

           0       0.88      1.00      0.94        15
           1       1.00      1.00      1.00        18
           2       1.00      0.88      0.94        17

    accuracy                           0.96        50
   macro avg       0.96      0.96      0.96        50
weighted avg       0.96      0.96      0.96        50
```

In [ ]:

In [ ]:

In [ ]:

In [ ]: