assignment-6

April 8, 2024

import libraries

```
[2]: import numpy as np
     import pandas as pd
     import seaborn as sns
```

import dataset

```
[3]: df=sns.load_dataset('iris')
```

understand the dataset

```
[4]: df.shape
```

```
[4]: (150, 5)
```

```
[5]: df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 150 entries, 0 to 149
Data columns (total 5 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   sepal_length  150 non-null    float64
 1   sepal_width   150 non-null    float64
 2   petal_length  150 non-null    float64
 3   petal_width   150 non-null    float64
 4   species       150 non-null    object
dtypes: float64(4), object(1)
memory usage: 6.0+ KB
```

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

```
[6]:    sepal_length  sepal_width  petal_length  petal_width species
     0           5.1          3.5           1.4          0.2  setosa
     1           4.9          3.0           1.4          0.2  setosa
     2           4.7          3.2           1.3          0.2  setosa
     3           4.6          3.1           1.5          0.2  setosa
     4           5.0          3.6           1.4          0.2  setosa
```

```
[7]: df.isnull().sum()
```

```
[7]: sepal_length    0
     sepal_width     0
     petal_length    0
     petal_width     0
     species         0
     dtype: int64
```

```
[8]: df.describe()
```

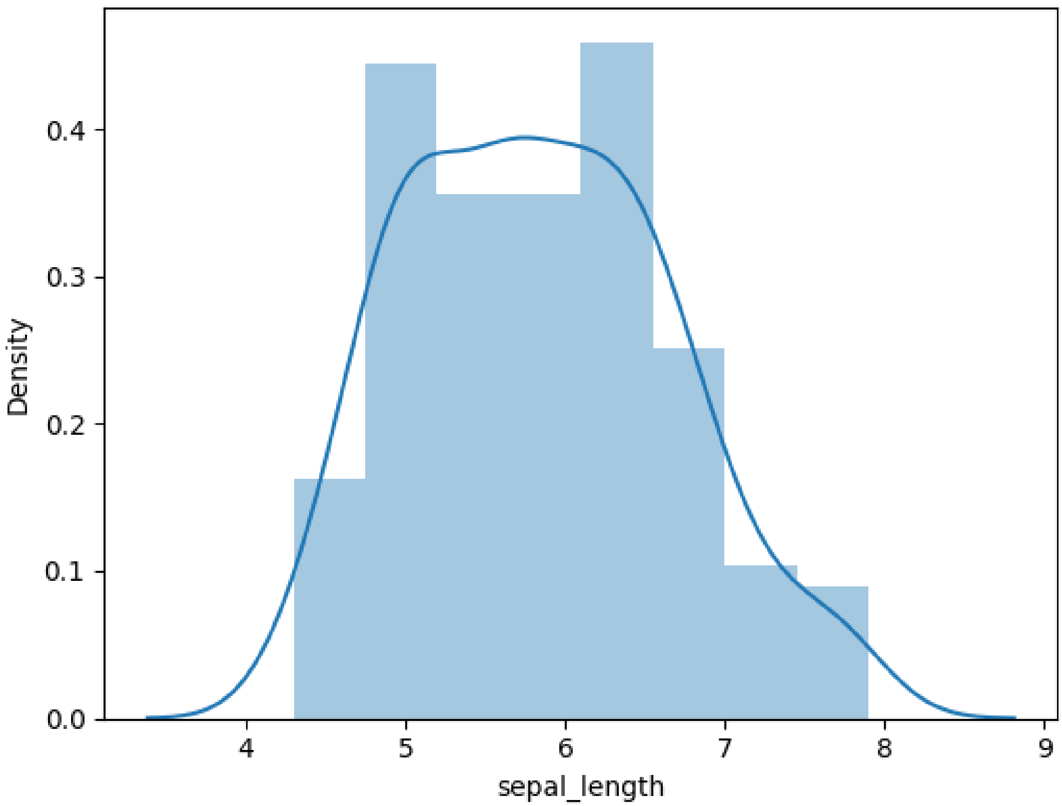
```
[8]:        sepal_length  sepal_width  petal_length  petal_width
     count    150.000000   150.000000    150.000000   150.000000
     mean       5.843333     3.057333      3.758000     1.199333
     std        0.828066     0.435866      1.765298     0.762238
     min        4.300000     2.000000      1.000000     0.100000
     25%        5.100000     2.800000      1.600000     0.300000
     50%        5.800000     3.000000      4.350000     1.300000
     75%        6.400000     3.300000      5.100000     1.800000
     max        7.900000     4.400000      6.900000     2.500000
```

exploratory data analysis

```
[11]: sns.distplot(df['sepal_length'])
```

```
<ipython-input-11-89138501e731>:1: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  sns.distplot(df['sepal_length'])
```

```
[11]: <Axes: xlabel='sepal_length', ylabel='Density'>
```



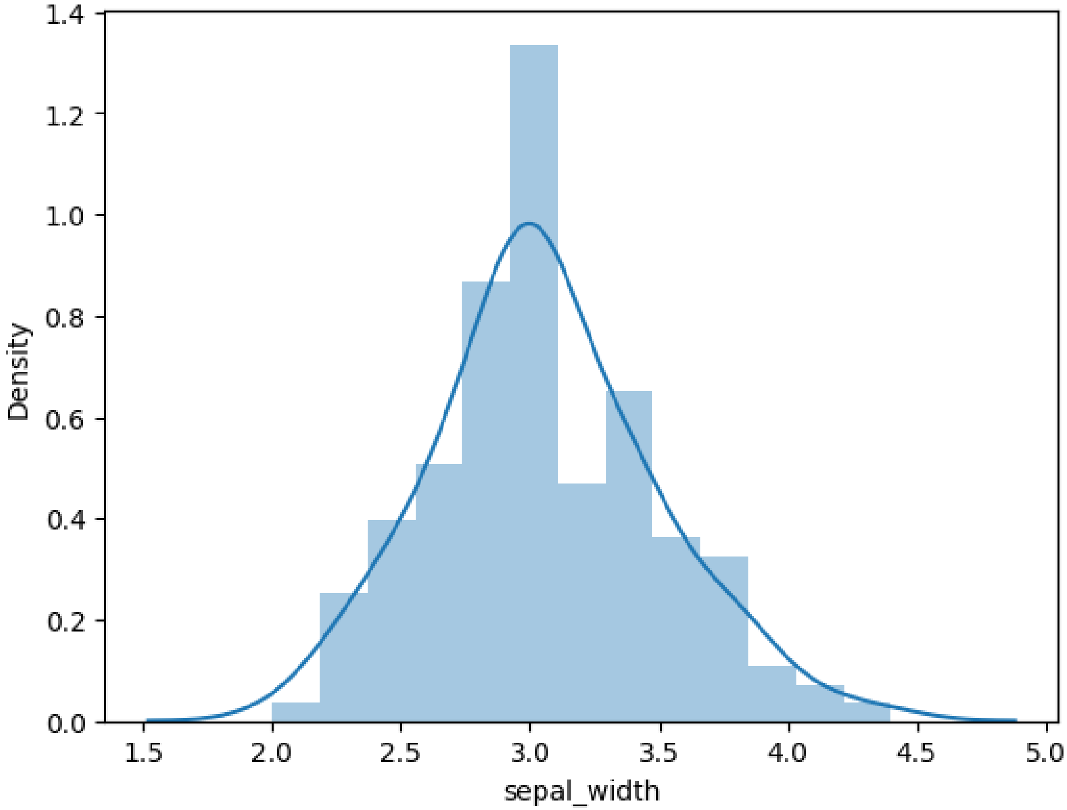
[12]:
```
df['sepal_length'].skew()
```

[12]:
```
0.3149109566369728
```

[13]:
```
sns.distplot(df['sepal_width'])
```

```
<ipython-input-13-6c237bf4ae06>:1: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  sns.distplot(df['sepal_width'])
```

[13]:
```
<Axes: xlabel='sepal_width', ylabel='Density'>
```



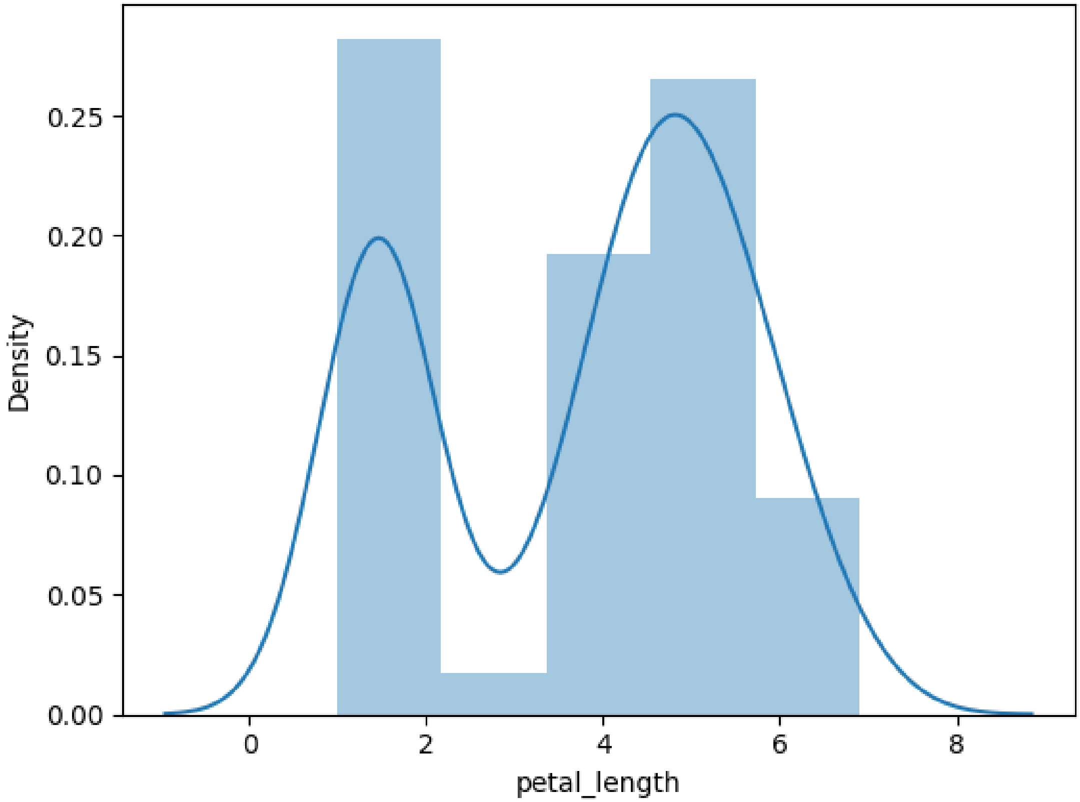
```
[14]: df['sepal_width'].skew()
```

```
[14]: 0.31896566471359966
```

```
[15]: sns.distplot(df['petal_length'])
```

```
<ipython-input-15-e0ee5e61121d>:1: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  sns.distplot(df['petal_length'])
```

```
[15]: <Axes: xlabel='petal_length', ylabel='Density'>
```



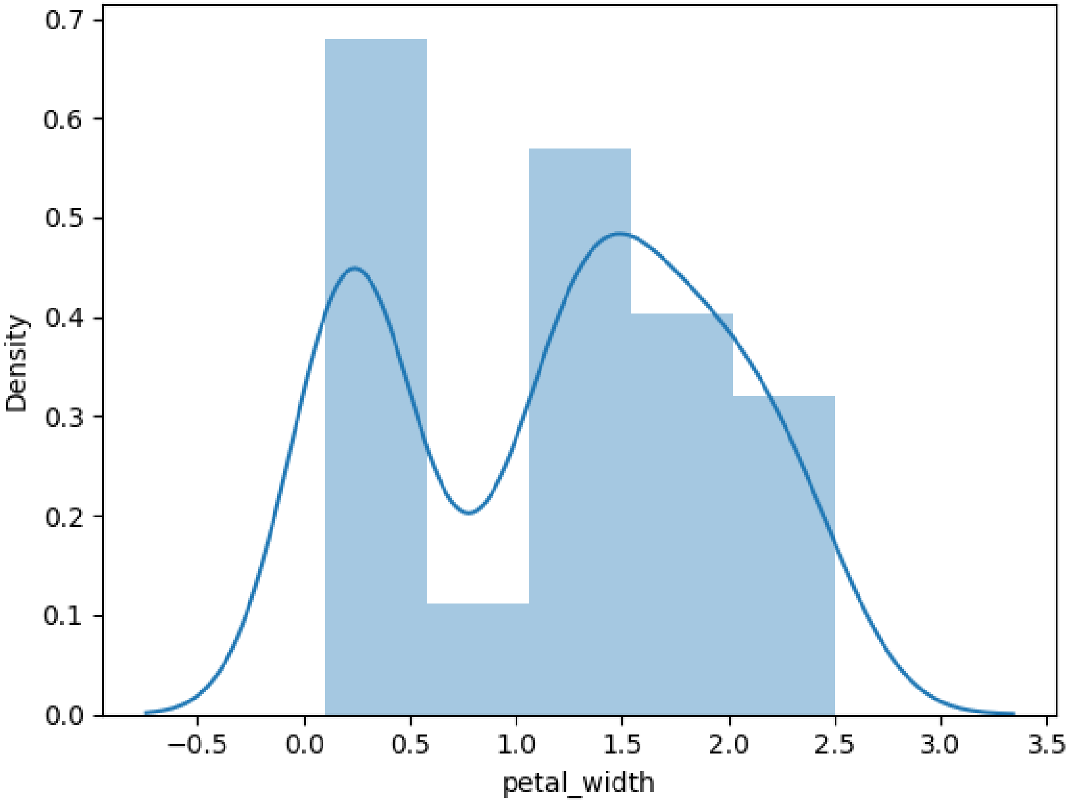
[16]:
```
df['petal_length'].skew()
```

[16]:
```
-0.27488417975101276
```

[17]:
```
sns.distplot(df['petal_width'])
```

```
<ipython-input-17-2b0be1ef4ca2>:1: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  sns.distplot(df['petal_width'])
```

[17]:
```
<Axes: xlabel='petal_width', ylabel='Density'>
```



```
[18]: df['petal_width'].skew()
```

```
[18]: -0.10296674764898116
```

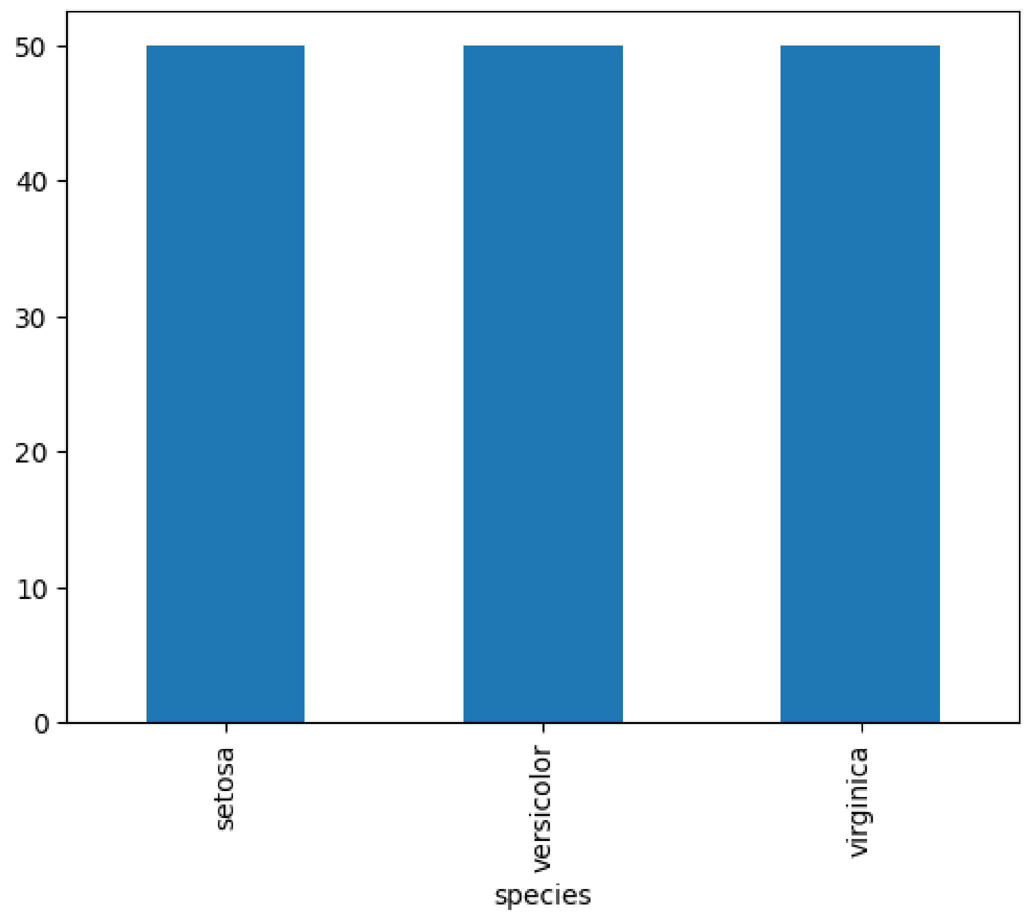
```
[19]: #all features are normally distributed(skewness value between -0.5 and 0.5)
```

```
[21]: df['species'].value_counts()
```

```
[21]: species
      setosa        50
      versicolor    50
      virginica     50
      Name: count, dtype: int64
```

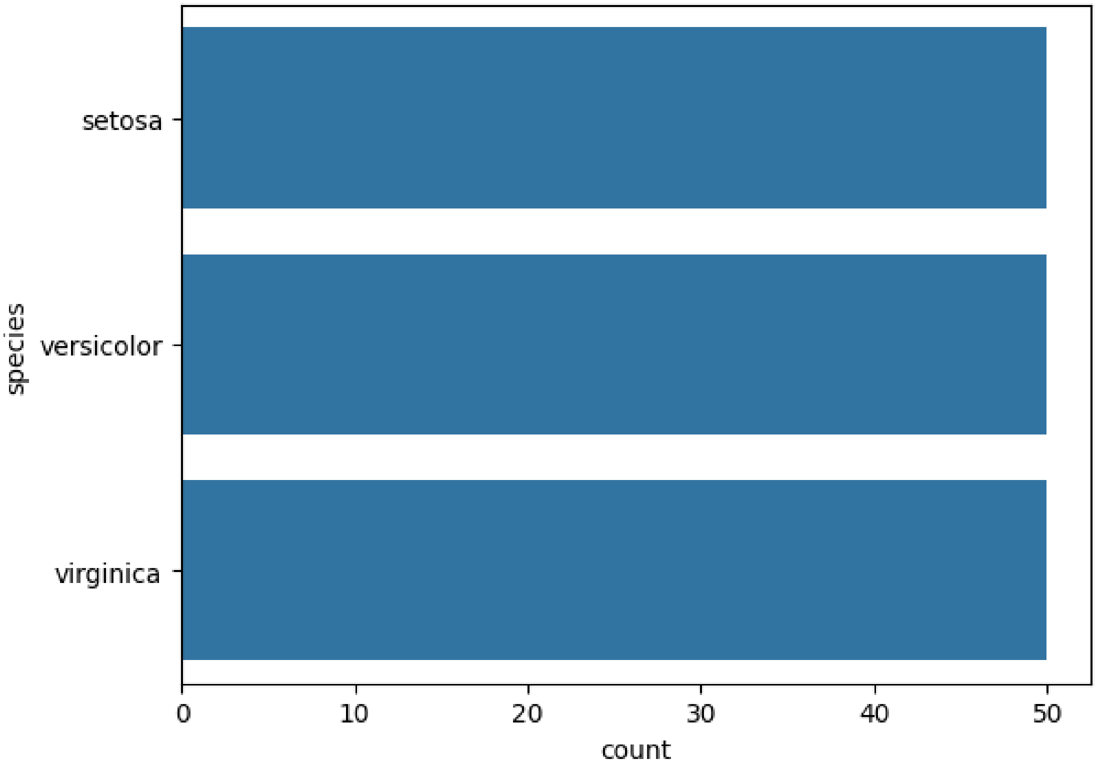
```
[22]: df['species'].value_counts().plot(kind='bar')
```

```
[22]: <Axes: xlabel='species'>
```



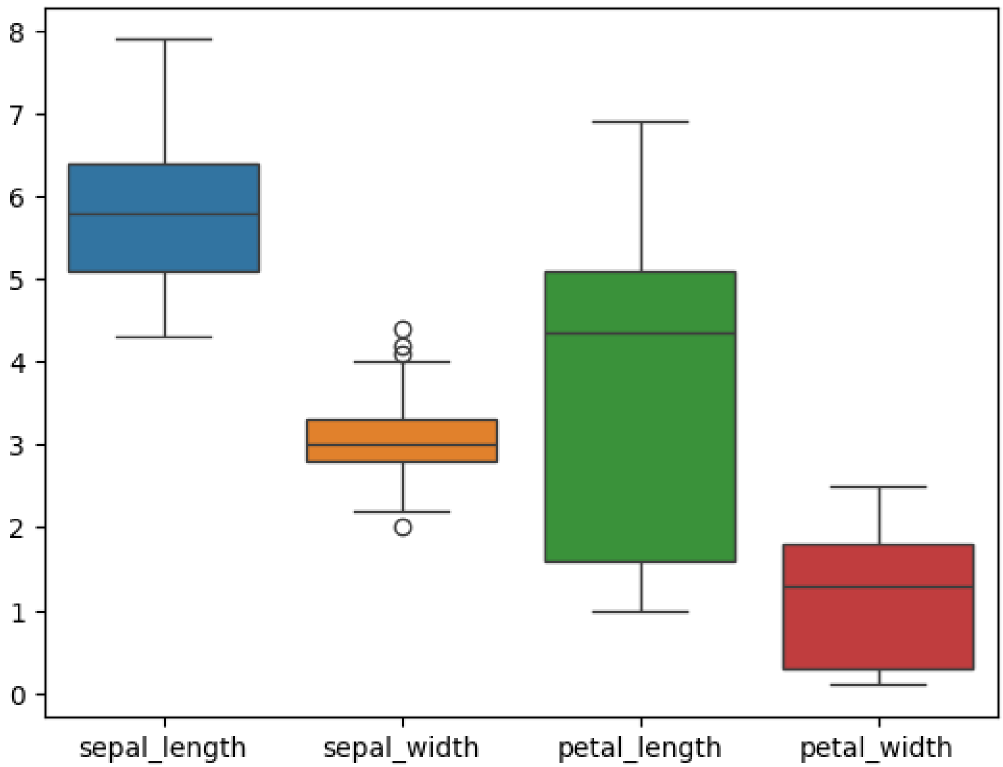
```
[26]: sns.countplot(df['species'])
```

```
[26]: <Axes: xlabel='count', ylabel='species'>
```



```
[29]: sns.boxplot(df)
```

```
[29]: <Axes: >
```



separate the dataset into x and y where y is the target variable

```
[30]: x=df.iloc[:,:-1]
```

```
[31]: x
```

```
[31]:      sepal_length  sepal_width  petal_length  petal_width
      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]
```

```
[32]: y=df.iloc[:,-1]
```

```
[33]: y
```

```
[33]: 0          setosa
      1          setosa
      2          setosa
      3          setosa
      4          setosa
                  …
      145    virginica
      146    virginica
      147    virginica
      148    virginica
      149    virginica
      Name: species, Length: 150, dtype: object
```

splitting dataset into training and testing

```
[34]: from sklearn.model_selection import train_test_split
```

```
[35]: x_train, x_test, y_train, y_test=train_test_split(x,y,train_size=0.
       ↪8,random_state=0)
```

```
[36]: x_train.shape
```

```
[36]: (120, 4)
```

```
[37]: x_test.shape
```

```
[37]: (30, 4)
```

```
[38]: y_train.shape
```

```
[38]: (120,)
```

```
[39]: y_test.shape
```

```
[39]: (30,)
```

model building

```
[41]: #build naive bayes classifier model
```

```
[40]: from sklearn.naive_bayes import GaussianNB
```

```
[45]: classifier=GaussianNB()
```

```
[47]: #train the model
      classifier.fit(x_train,y_train)
```

[47]:

```
GaussianNB()
```

[48]:

```
#testing the model
y_pred=classifier.predict(x_test)
```

[52]:

```
result=pd.DataFrame({'actual':y_test,'prediction':y_pred})
```

[53]:

```
result
```

[53]:

```
         actual  prediction
114   virginica   virginica
62   versicolor  versicolor
33       setosa      setosa
107   virginica   virginica
7        setosa      setosa
100   virginica   virginica
40       setosa      setosa
86   versicolor  versicolor
76   versicolor  versicolor
71   versicolor  versicolor
134   virginica  versicolor
51   versicolor  versicolor
73   versicolor  versicolor
54   versicolor  versicolor
63   versicolor  versicolor
37       setosa      setosa
78   versicolor  versicolor
90   versicolor  versicolor
45       setosa      setosa
16       setosa      setosa
121   virginica   virginica
66   versicolor  versicolor
24       setosa      setosa
8        setosa      setosa
126   virginica   virginica
22       setosa      setosa
44       setosa      setosa
97   versicolor  versicolor
93   versicolor  versicolor
26       setosa      setosa
```

evaluate the model

[59]:

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

[61]:

```
cm=confusion_matrix(y_test,y_pred)
```

```
[62]: cm
```

```
[62]: array([[11,  0,  0],
              [ 0, 13,  0],
              [ 0,  1,  5]])
```

```
[63]: cm=confusion_matrix(y_pred,y_test)
```

```
[64]: cm
```

```
[64]: array([[11,  0,  0],
              [ 0, 13,  1],
              [ 0,  0,  5]])
```

```
[65]: accuracy_score(y_test,y_pred)
```

```
[65]: 0.9666666666666667
```

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

```
[81]: 0.9666666666666667
```

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

```
[80]: 0.9666666666666667
```

```
[79]: fbeta_score(y_test,y_pred,beta=0.5,average='micro')
```

```
[79]: 0.9666666666666666
```

```
[82]: print(classification_report(y_test,y_pred))
```

```
              precision    recall  f1-score   support

      setosa       1.00      1.00      1.00        11
  versicolor       0.93      1.00      0.96        13
   virginica       1.00      0.83      0.91         6

    accuracy                           0.97        30
   macro avg       0.98      0.94      0.96        30
weighted avg       0.97      0.97      0.97        30
```

```
[ ]:
```