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

In [3]:
```
dt=pd.read_csv("iris_dataset.csv")
```

In [4]:
```
dt.head(10)
```

Out[4]:

	sepal_length	sepal_width	petal_length	petal_width	target
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
5	5.4	3.9	1.7	0.4	Iris-setosa
6	4.6	3.4	1.4	0.3	Iris-setosa
7	5.0	3.4	1.5	0.2	Iris-setosa
8	4.4	2.9	1.4	0.2	Iris-setosa
9	4.9	3.1	1.5	0.1	Iris-setosa

In [5]:
```
dt.info()
```

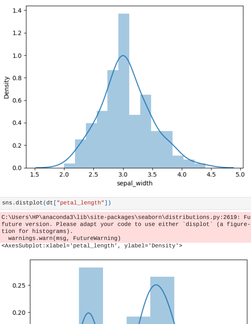
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 150 entries, 0 to 149
Data columns (total 5 columns):
 #   Column        Non-Null Count  Dtype  
---  ------        --------------  -----  
 0   sepal_length  150 non-null    float64
 1   sepal_width   150 non-null    float64
 2   petal_length  150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

Univariate Analysis

In [6]:
```
sns.distplot(dt["sepal_length"])
```

```
C:\Users\HP\anaconda3\lib\site-packages\seaborn\distributions.py:2619: FutureWarning: `distplot` is a deprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```

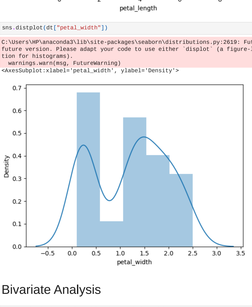
Out[6]:
```
<AxesSubplot:xlabel='sepal_length', ylabel='Density'>
```



In [7]:
```
sns.distplot(dt["sepal_width"])
```

```
C:\Users\HP\anaconda3\lib\site-packages\seaborn\distributions.py:2619: FutureWarning: `distplot` is a deprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```

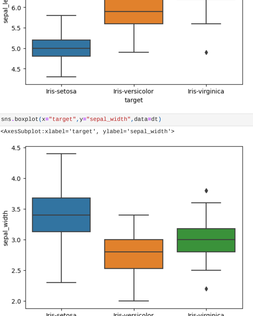
Out[7]:
```
<AxesSubplot:xlabel='sepal_width', ylabel='Density'>
```



In [8]:
```
sns.distplot(dt["petal_length"])
```

```
C:\Users\HP\anaconda3\lib\site-packages\seaborn\distributions.py:2619: FutureWarning: `distplot` is a deprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```

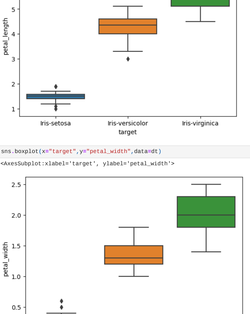
Out[8]:
```
<AxesSubplot:xlabel='petal_length', ylabel='Density'>
```



In [9]:
```
sns.distplot(dt["petal_width"])
```

```
C:\Users\HP\anaconda3\lib\site-packages\seaborn\distributions.py:2619: FutureWarning: `distplot` is a deprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```

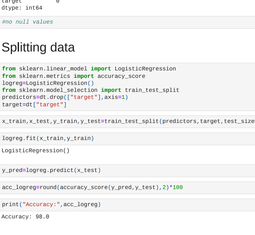
Out[9]:
```
<AxesSubplot:xlabel='petal_width', ylabel='Density'>
```



Bivariate Analysis

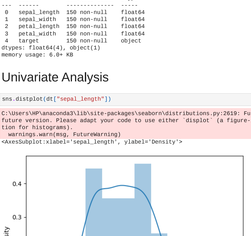
In [12]:
```
sns.boxplot(y="sepal_length",x="target",data=dt)
```

Out[12]:
```
<AxesSubplot:xlabel='target', ylabel='sepal_length'>
```



In [13]:
```
sns.boxplot(x="target",y="sepal_width",data=dt)
```

Out[13]:
```
<AxesSubplot:xlabel='target', ylabel='sepal_width'>
```



In [14]:
```
sns.boxplot(x="target",y="petal_length",data=dt)
```

Out[14]:
```
<AxesSubplot:xlabel='target', ylabel='petal_length'>
```


In [15]:
```
sns.boxplot(x="target",y="petal_width",data=dt)
```

Out[15]:
```
<AxesSubplot:xlabel='target', ylabel='petal_width'>
```


In [16]:
```
dt.isnull().sum()
```

Out[16]:
```
sepal_length    0
sepal_width     0
petal_length    0
petal_width     0
target          0
dtype: int64
```

In [17]:
```
#no null values
```

Splitting data

In [23]:
```
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score
logreg=LogisticRegression()
from sklearn.model_selection import train_test_split
predictors=dt.drop(["target"],axis=1)
target=dt["target"]
```

In [24]:
```
x_train,x_test,y_train,y_test=train_test_split(predictors,target,test_size=0.3,random_state=0)
```

In [25]:
```
logreg.fit(x_train,y_train)
```

Out[25]:
```
LogisticRegression()
```

In [26]:
```
y_pred=logreg.predict(x_test)
```

In [27]:
```
acc_logreg=round(accuracy_score(y_pred,y_test),2)*100
```

In [28]:
```
print("Accuracy:",acc_logreg)
```

```
Accuracy: 98.0
```