In [2]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sb
from sklearn.linear_model import LogisticRegression
```

In [4]:

```
iris_data=pd.read_csv("C:\\Users\\racha\\OneDrive\\Desktop\\Datascience\\iris flower.zip
iris_data.head(10)
```

Out[4]:

	sepal_length	sepal_width	petal_length	petal_width	species
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]:

```
iris_data.describe()
```

Out[5]:

	sepal_length	sepal_width	petal_length	petal_width
count	150.000000	150.000000	150.000000	150.000000
mean	5.843333	3.054000	3.758667	1.198667
std	0.828066	0.433594	1.764420	0.763161
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

In [6]:

```
iris_data.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
```

In [7]:

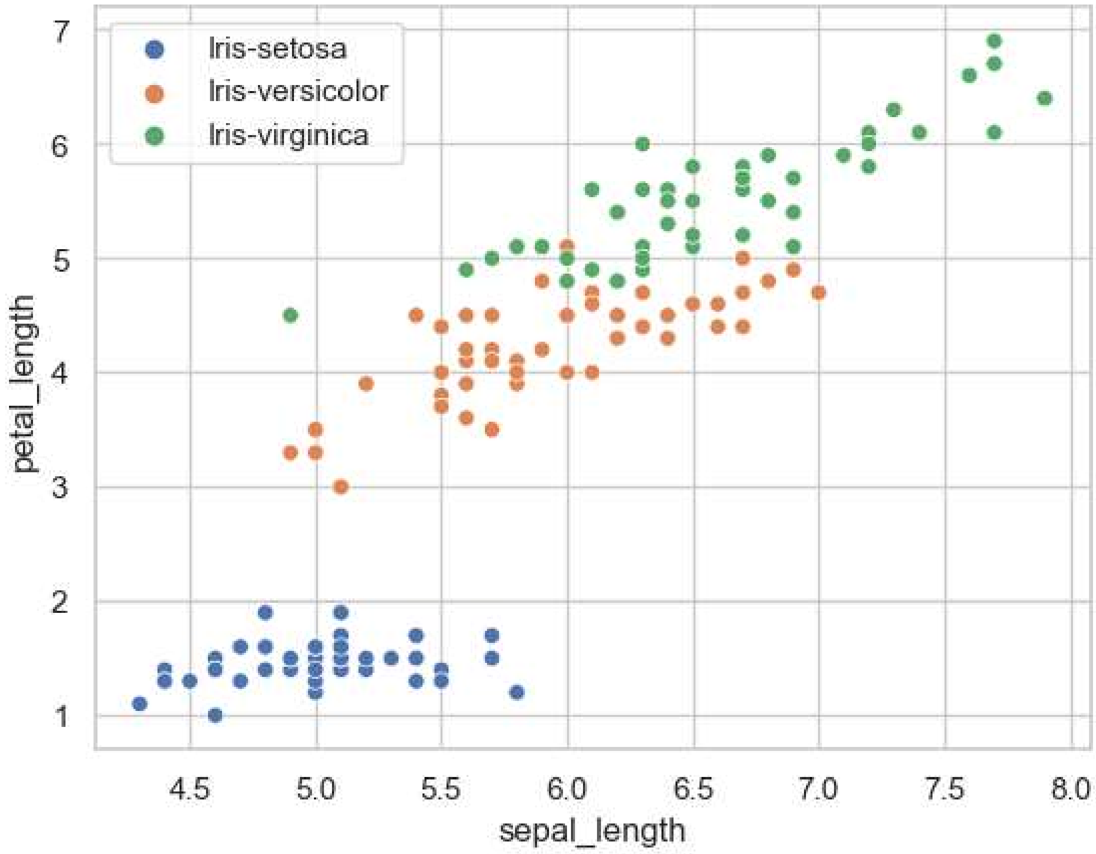
```
iris_data['species'].value_counts()
```

Out[7]:

```
Iris-setosa        50
Iris-versicolor    50
Iris-virginica     50
Name: species, dtype: int64
```

In [16]:

```
sb.set(style="whitegrid")
sb.scatterplot(data=iris_data, x="sepal_length", y="petal_length", hue="species")
plt.legend()
plt.show()
```



In [17]:

```
x=iris_data[["sepal_length","sepal_width","petal_length","petal_width"]].values
y=iris_data[["species"]].values
```

In [19]:

```
model=LogisticRegression()
model.fit(x,y)
```

```
C:\Users\racha\anaconda3\lib\site-packages\sklearn\utils\validation.py:99
3: DataConversionWarning: A column-vector y was passed when a 1d array was
expected. Please change the shape of y to (n_samples, ), for example using
ravel().
  y = column_or_1d(y, warn=True)
```

Out[19]:

```
LogisticRegression()
```

In [22]:

```
model.score(x,y).round(4)
```

Out[22]:

```
0.9733
```

In [24]:

```
actual=y
predicted=model.predict(x)
```

In [25]:

```
from sklearn import metrics
print(metrics.classification_report(actual,predicted))
```

```
                 precision    recall  f1-score   support

    Iris-setosa       1.00      1.00      1.00        50
Iris-versicolor       0.98      0.94      0.96        50
 Iris-virginica       0.94      0.98      0.96        50

       accuracy                           0.97       150
      macro avg       0.97      0.97      0.97       150
   weighted avg       0.97      0.97      0.97       150
```

In [26]:

```
print(metrics.confusion_matrix(actual,predicted))
```

```
[[50  0  0]
 [ 0 47  3]
 [ 0  1 49]]
```

In [27]:

```
predicted=model.predict([[5.1,3.5,1.4,0.2]])
predicted
```

Out[27]:

```
array(['Iris-setosa'], dtype=object)
```

In [29]:

```
predicted=model.predict([[6.5,2.8,2.2,0.5]])
predicted
```

Out[29]:

```
array(['Iris-versicolor'], dtype=object)
```