In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sb
from sklearn.linear_model import LogisticRegression
```

In [6]:
```
iris_data=pd.read_csv("C:\\Users\\SRI KAAVYA\\OneDrive\\Desktop\\Internship pr
iris_data.tail(10)
```

Out[6]:

	sepal_length	sepal_width	petal_length	petal_width	species
140	6.7	3.1	5.6	2.4	Iris-virginica
141	6.9	3.1	5.1	2.3	Iris-virginica
142	5.8	2.7	5.1	1.9	Iris-virginica
143	6.8	3.2	5.9	2.3	Iris-virginica
144	6.7	3.3	5.7	2.5	Iris-virginica
145	6.7	3.0	5.2	2.3	Iris-virginica
146	6.3	2.5	5.0	1.9	Iris-virginica
147	6.5	3.0	5.2	2.0	Iris-virginica
148	6.2	3.4	5.4	2.3	Iris-virginica
149	5.9	3.0	5.1	1.8	Iris-virginica

In [7]:
```
iris_data.describe()
```

Out[7]:

	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

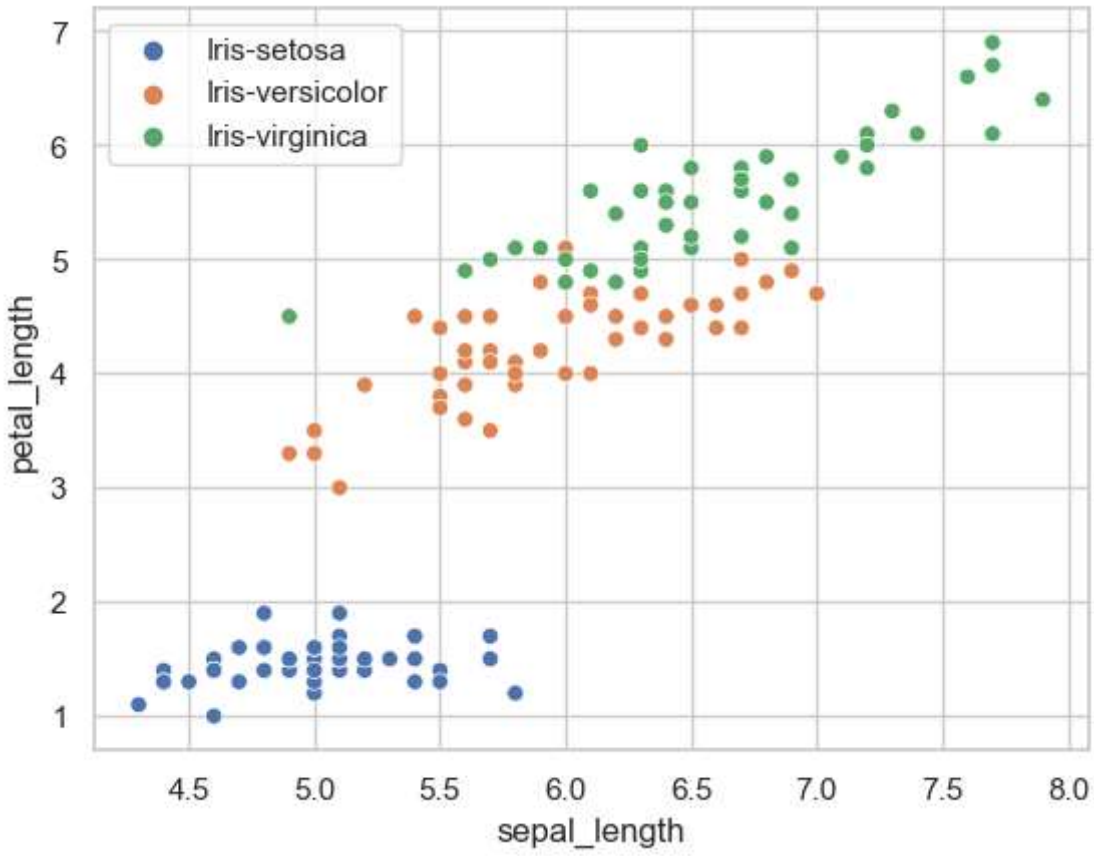
In [8]:
```
iris_data.info()
```

```
<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 [10]:
```
iris_data['species'].value_counts()
```

Out[10]:
```
Iris-setosa        50
Iris-versicolor    50
Iris-virginica     50
Name: species, dtype: int64
```

In [11]:
```
sb.set(style="whitegrid")
sb.scatterplot(data=iris_data,x="sepal_length",y="petal_length",hue="species")
plt.legend()
plt.show()
```



In [12]:
```
x=iris_data[["sepal_length","sepal_width","petal_length","petal_width"]].value
y=iris_data[["species"]].values
```

In [13]:
```
model=LogisticRegression()
```

In [14]:
```
model.fit(x,y)
```

```
C:\Users\SRI KAAVYA\anaconda3\Lib\site-packages\sklearn\utils\validation.py:1
143: 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 ra
vel().
  y = column_or_1d(y, warn=True)
```

Out[14]:
```
LogisticRegression()
```
In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.
On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.

In [15]:
```
model.score(x,y).round(4)
```

Out[15]:
```
0.9733
```

In [16]:
```
actual=y
predicted=model.predict(x)
```

In [17]:
```
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 [18]:
```
print(metrics.confusion_matrix(actual,predicted))
```

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

In [19]:
```
predicted=model.predict([[5.1,3.5,1.4,0.2]])
predicted
```

Out[19]:
```
array(['Iris-setosa'], dtype=object)
```

In [20]:
```
predicted=model.predict([[6.5,2.8,2.2,0.5]])
predicted
```

Out[20]:
```
array(['Iris-versicolor'], dtype=object)
```