In [2]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
from sklearn.metrics import accuracy_score
import warnings
warnings.filterwarnings('ignore')
```

In [4]:

```
df = pd.read_csv(r'C:\Users\HP\Downloads\IRIS.csv')
```

In [5]:

```
df
```

Out[5]:

```
     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
...           ...          ...           ...          ...             ...
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

150 rows × 5 columns
```

In [6]:

```
df.isna().sum()
```

Out[6]:

```
sepal_length    0
sepal_width     0
petal_length    0
petal_width     0
species         0
dtype: int64
```

In [7]:

```
df.duplicated().sum()
```

Out[7]:

```
3
```

In [8]:

```
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
```

In [9]:

```
df.describe()
```

Out[9]:

	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 [10]:

```
df.drop_duplicates(inplace=True)
```

In [11]:

```
df
```

Out[11]:

	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
...	...	...	...	...	...
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

147 rows × 5 columns

In [12]:

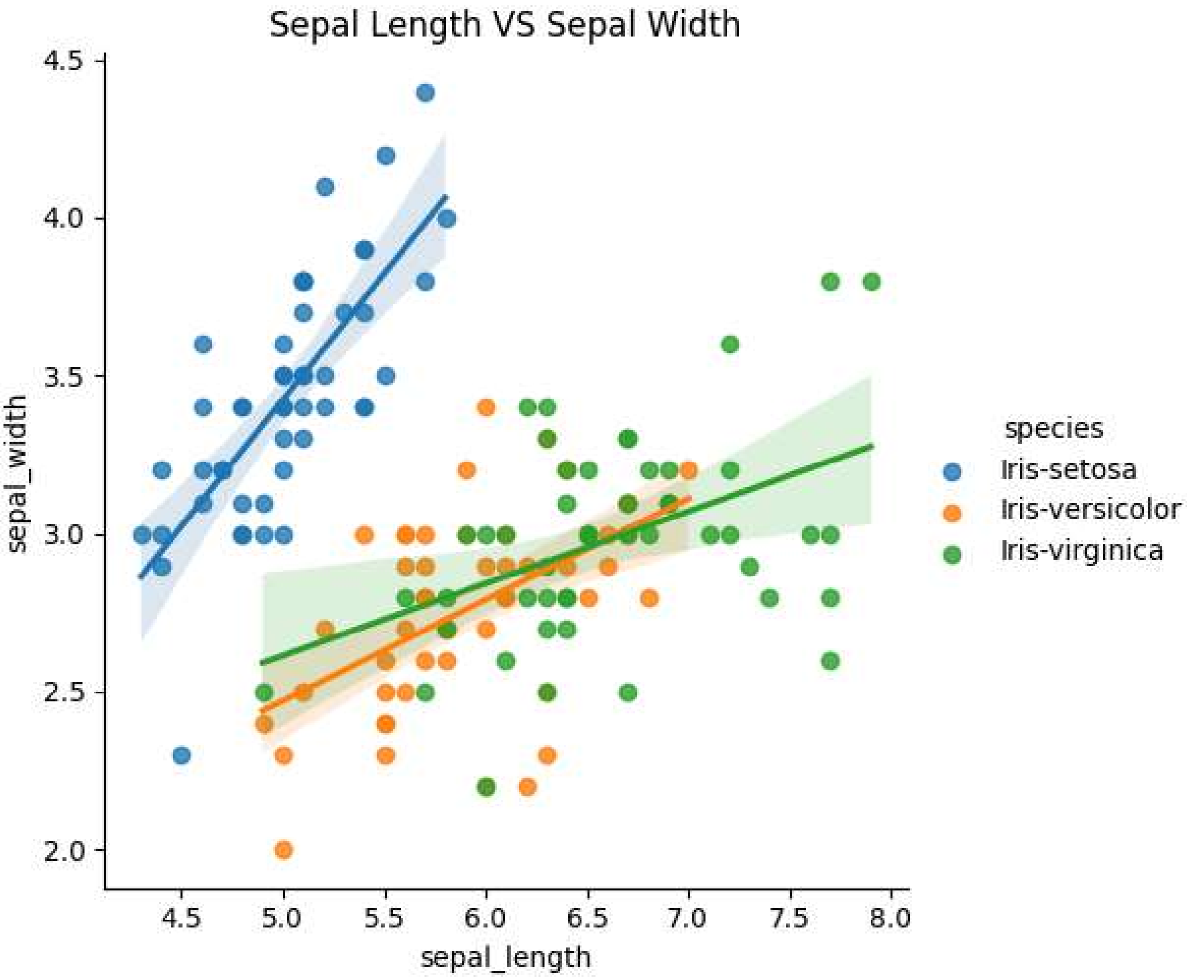
```
df['species'].value_counts()
```

Out[12]:

```
species
Iris-versicolor    50
Iris-virginica     49
Iris-setosa        48
Name: count, dtype: int64
```

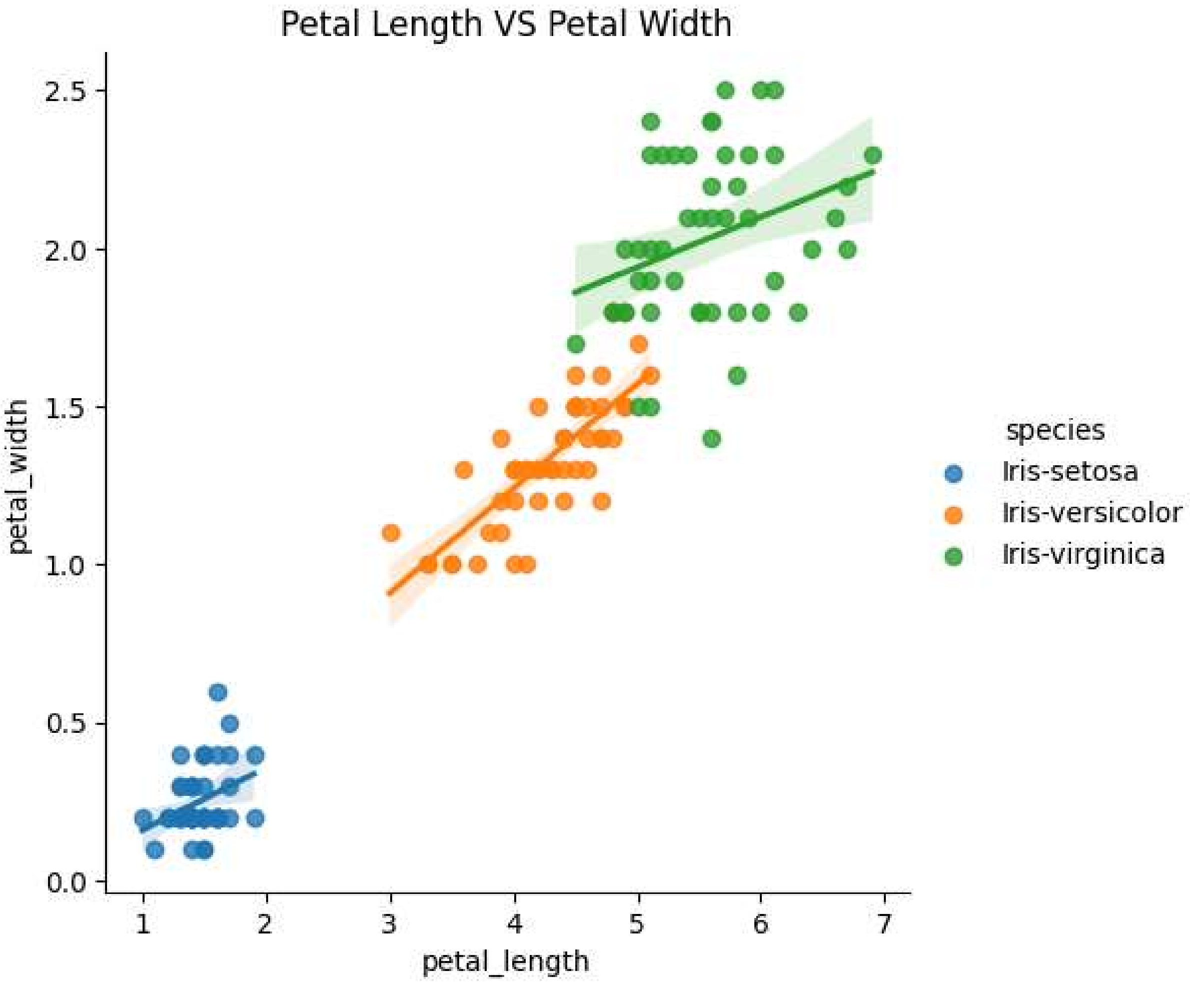
In [13]:

```
sns.lmplot(
    x="sepal_length",
    y="sepal_width",
    hue="species",
    data=df
)

plt.title("Sepal Length VS Sepal Width")
plt.show()
```



In [14]:

```
sns.lmplot(
    x="petal_length",
    y="petal_width",
    hue="species",
    data=df
)

plt.title("Petal Length VS Petal Width")
plt.show()
```



```
In [15]: label_encoder = LabelEncoder()
         df['species'] = label_encoder.fit_transform(df['species'])
```

```
In [16]: df
```

Out[16]:

	sepal_length	sepal_width	petal_length	petal_width	species
0	5.1	3.5	1.4	0.2	0
1	4.9	3.0	1.4	0.2	0
2	4.7	3.2	1.3	0.2	0
3	4.6	3.1	1.5	0.2	0
4	5.0	3.6	1.4	0.2	0
...	...	...	...	...	...
145	6.7	3.0	5.2	2.3	2
146	6.3	2.5	5.0	1.9	2
147	6.5	3.0	5.2	2.0	2
148	6.2	3.4	5.4	2.3	2
149	5.9	3.0	5.1	1.8	2

147 rows × 5 columns

In [19]:

```
x = df.drop(columns='species')
y = df.species
```

In [18]:

```
x
```

Out[18]:

	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

147 rows × 4 columns

In [20]:

```
y
```

Out[20]:

```
0      0
1      0
2      0
3      0
4      0
      ..
145    2
146    2
147    2
148    2
149    2
Name: species, Length: 147, dtype: int32
```

In [21]:

```
X_train, x_test, Y_train, y_test = train_test_split(x, y, test_size=0.2, random_sta
```

In [24]:

```
from sklearn.linear_model import LogisticRegression

# Instantiate the model
model = LogisticRegression()

# Fit the model with training data
model.fit(X_train, Y_train)
```

Out[24]:

```
▾ LogisticRegression ⓘ ?

LogisticRegression()
```

In [25]:

```
model.score(x_test, y_test)
```

Out[25]:

```
1.0
```

In [26]:

```
model.score(X_train, Y_train)
```

Out[26]:

```
0.9743589743589743
```

In [27]:

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

Out[27]:

```
array([0])
```

In [28]:

```
y_predicted = model.predict(x_test)
```

In [29]:

```
y_predicted
```

Out[29]:

```
array([2, 1, 1, 1, 0, 0, 1, 0, 2, 2, 0, 2, 0, 2, 1, 2, 2, 0, 1, 1, 0, 2,
       1, 0, 2, 0, 0, 2, 2, 0])
```

In [30]:

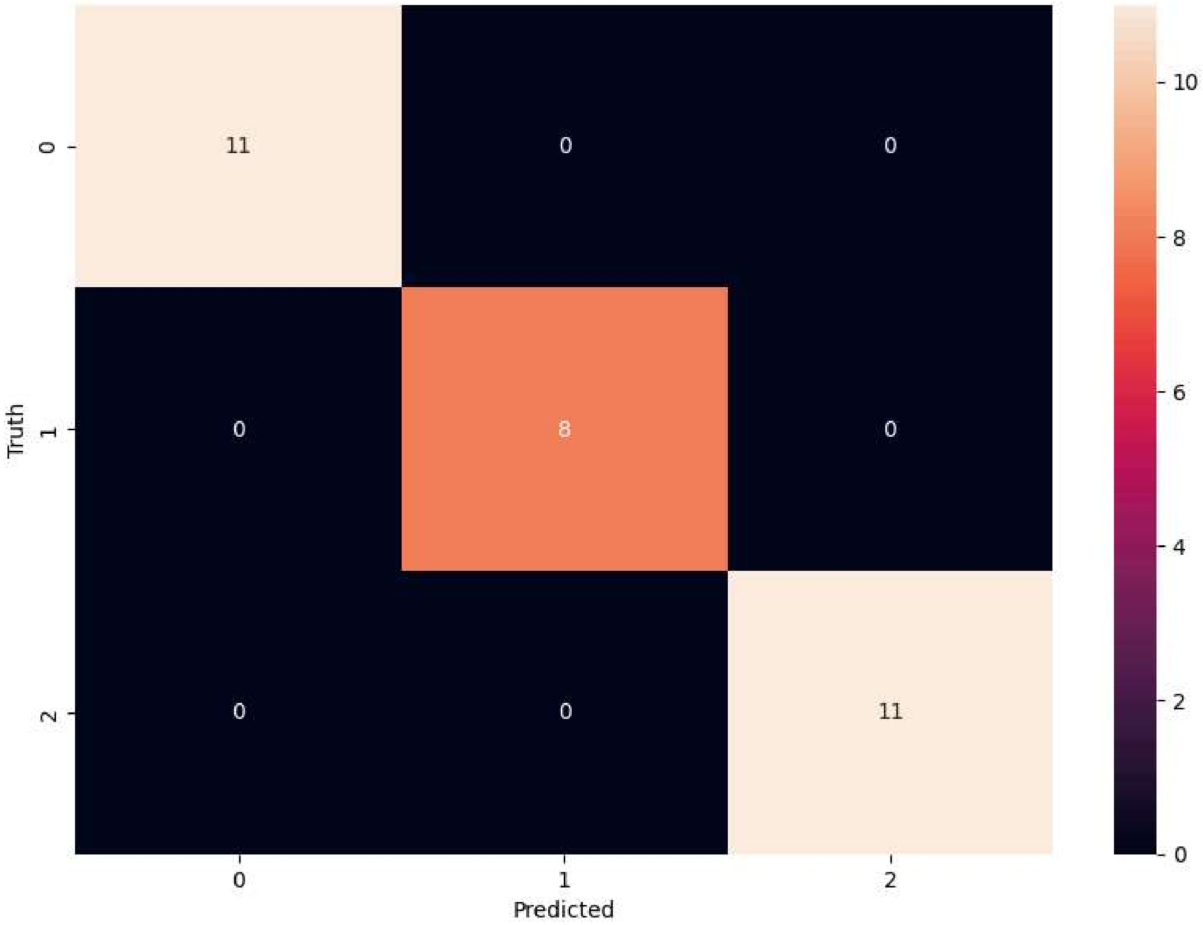
```
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y_test, y_predicted) # compare between predicted values, actu
cm
```

Out[30]:

```
array([[11,  0,  0],
       [ 0,  8,  0],
       [ 0,  0, 11]], dtype=int64)
```

In [31]:
```python
import seaborn as sn
plt.figure(figsize = (10,7))
sn.heatmap(cm, annot = True)
plt.xlabel('Predicted')
plt.ylabel('Truth')
```

Out[31]:
```
Text(95.72222222222221, 0.5, 'Truth')
```



In [ ]: