In [98]:
```
import pandas as pd
import numpy as np
from matplotlib import pyplot as plt
import seaborn as sns
```

In [99]:
```
df=pd.read_csv("IRIS.csv")
```

In [100]:
```
df.head()
```

Out[100]:

	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

In [101]:
```
df.tail()
```

Out[101]:

	sepal_length	sepal_width	petal_length	petal_width	species
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 [102]:
```
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 [103]:
```
df.describe()
```

Out[103]:

	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 [104]:
```
df.isnull().sum()
```

Out[104]:
```
sepal_length    0
sepal_width     0
petal_length    0
petal_width     0
species         0
dtype: int64
```

In [105]:
```
df.duplicated()
```

Out[105]:
```
0      False
1      False
2      False
3      False
4      False
       ...
145    False
146    False
147    False
148    False
149    False
Length: 150, dtype: bool
```

In [106]:
```
df[df.duplicated()]
```

Out[106]:

	sepal_length	sepal_width	petal_length	petal_width	species
34	4.9	3.1	1.5	0.1	Iris-setosa
37	4.9	3.1	1.5	0.1	Iris-setosa
142	5.8	2.7	5.1	1.9	Iris-virginica

In [107]:
```
df.drop_duplicates()
```

Out[107]:

	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 [108]:
```
df.columns
```

Out[108]:
```
Index(['sepal_length', 'sepal_width', 'petal_length', 'petal_width',
       'species'],
      dtype='object')
```

In [109]:
```
df.describe(include=['O']).T
```

Out[109]:

	count	unique	top	freq
species	150	3	Iris-setosa	50

In [110]:
```
df['species'].unique()
```

Out[110]:
```
array(['Iris-setosa', 'Iris-versicolor', 'Iris-virginica'], dtype=object)
```

In [111]:
```
df['species'].replace(['Iris-setosa','Iris-virginica','Iris-versicolor'],[0,1,2
print(df)
```

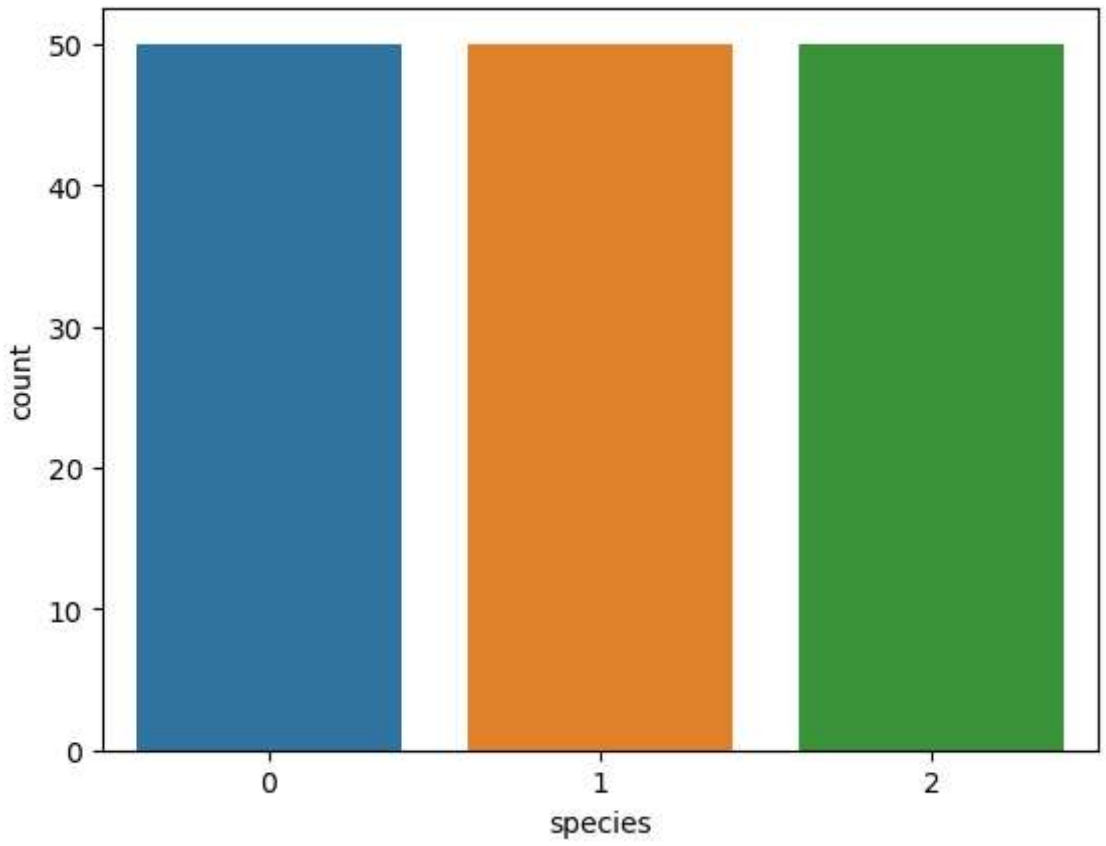
```
     sepal_length  sepal_width  petal_length  petal_width  species
0             5.1          3.5           1.4          0.2        0
1             4.9          3.0           1.4          0.2        0
2             4.7          3.2           1.3          0.2        0
3             4.6          3.1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

[150 rows x 5 columns]
```

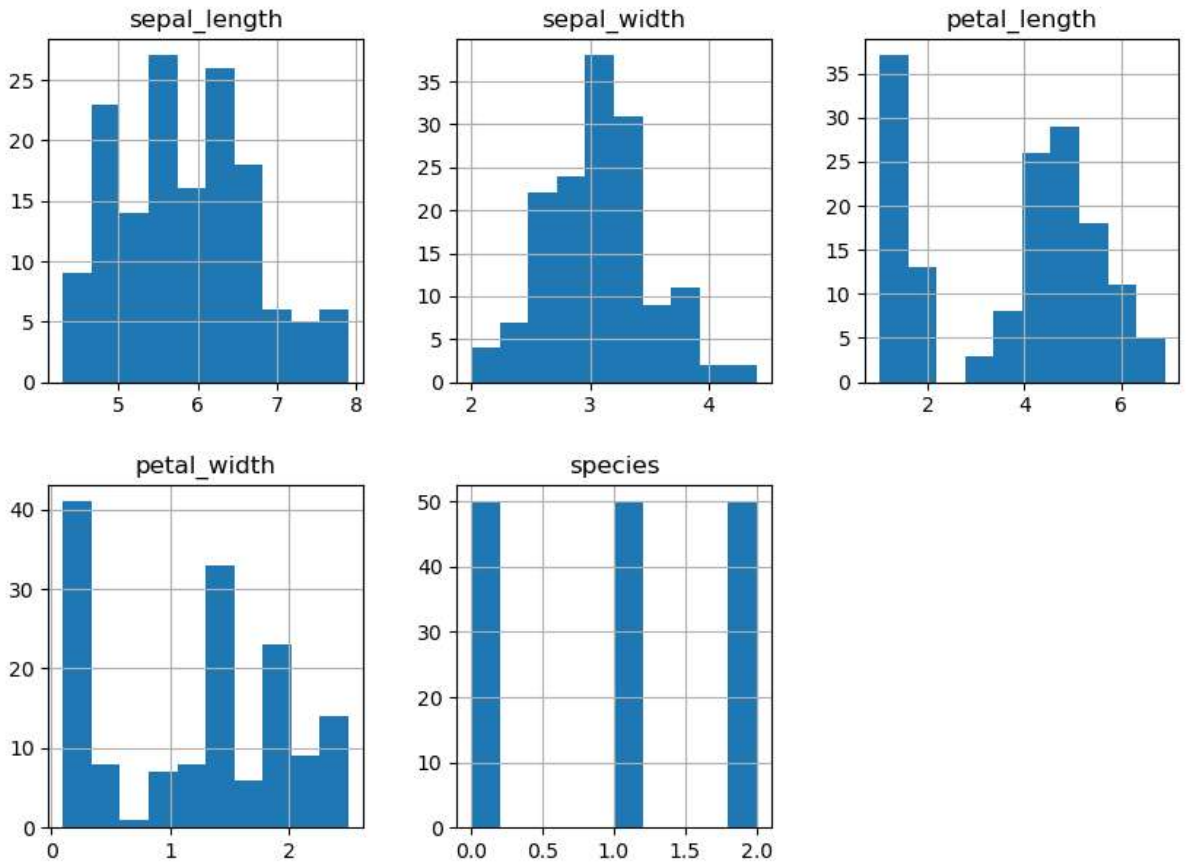
In [112]:
```
print("Iris-setosa=0  Iris-virginica=1  Iris-versicolor=2")
sns.countplot(x='species',data=df)
```

```
Iris-setosa=0  Iris-virginica=1  Iris-versicolor=2
```

Out[112]:
```
<Axes: xlabel='species', ylabel='count'>
```



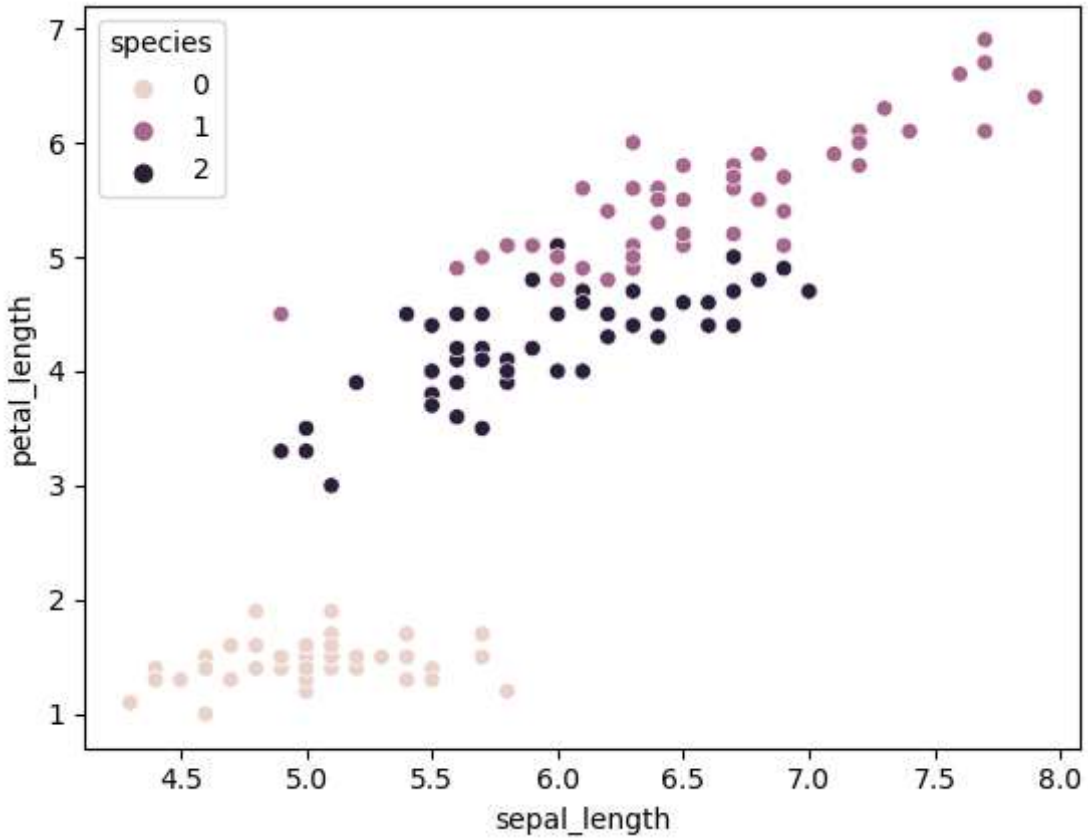
In [45]:
```
df.hist(layout=(2,3),figsize=(10,7))
plt.show()
```



In [113]:
```
print("Iris-setosa=0  Iris-virginica=1  Iris-versicolor=2")
sns.scatterplot(x="sepal_length",y="petal_length",hue="species",data=df)
```

```
Iris-setosa=0  Iris-virginica=1  Iris-versicolor=2
```

Out[113]:
```
<Axes: xlabel='sepal_length', ylabel='petal_length'>
```



In [46]:
```
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
```

In [84]:
```
x=df.drop('species',axis=1)
y=df['species']
```

In [85]:
```
model_df={}
def model_val(model,x,y):
    x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.5,random_sta
    model.fit(x_train,y_train)
    pred=model.predict(x_test)
    print(f"{model} accuracy is {accuracy_score(y_test,pred)}")
```

In [89]:
```
#Linear regression
from sklearn.linear_model import LogisticRegression
model=LogisticRegression()
model_val(model,x,y)
```

```
LogisticRegression() accuracy is 1.0
```

In [90]:
```
#SVM
from sklearn import svm
model=svm.SVC()
model_val(model,x,y)
```

```
SVC() accuracy is 0.9866666666666667
```

In [91]:
```
#Decision Tree
from sklearn.tree import DecisionTreeClassifier
model=DecisionTreeClassifier()
model_val(model,x,y)
```

```
DecisionTreeClassifier() accuracy is 0.9866666666666667
```

In [ ]:
```

```