

In [1]:

```
from sklearn.svm import SVC
import pandas as pd
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
```

In [2]:

```
data = pd.read_csv('iris.csv')
data.head()
```

Out[2]:

	sepal_length	sepal_width	petal_length	petal_width	target
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

In [3]:

```
data.shape
```

Out[3]:

```
(150, 5)
```

In [4]:

```
data.target.unique()
```

Out[4]:

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

In [5]:

```
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   target         150 non-null   int64   
dtypes: float64(4), int64(1)
memory usage: 6.0 KB
```

In [6]:

```
data.describe()
```

Out[6]:

	sepal_length	sepal_width	petal_length	petal_width	target
count	150.000000	150.000000	150.000000	150.000000	150.000000
mean	5.843333	3.054000	3.758667	1.198667	1.000000

std	sepal_length	sepal_width	petal_length	petal_width	target
	0.828066	0.433594	1.764420	0.763161	0.819232
min	4.300000	2.000000	1.000000	0.100000	0.000000
25%	5.100000	2.800000	1.600000	0.300000	0.000000
50%	5.800000	3.000000	4.350000	1.300000	1.000000
75%	6.400000	3.300000	5.100000	1.800000	2.000000
max	7.900000	4.400000	6.900000	2.500000	2.000000

In [7]:

```
data.isnull().sum()
```

Out[7]:

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

In [8]:

```
from collections import Counter
Counter(data.target)
```

Out[8]:

```
Counter({0: 50, 1: 50, 2: 50})
```

In [9]:

```
data.target.value_counts()
```

Out[9]:

```
target
0      50
1      50
2      50
Name: count, dtype: int64
```

In [10]:

```
X = data.iloc[:, :-1]
y = data[['target']]
```

In [11]:

```
X.head()
```

Out[11]:

	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

In [12]:

```
y.head()
```

Out[12]:

target	
0	0
1	0
2	0
3	0
4	0

In [13]:

```
X_train,X_test,y_train,y_test = train_test_split(X,y,random_state=10,test_size=0.5)
```

In [14]:

```
model = SVC()
model.fit(X_train,y_train)
y_predict = model.predict(X_test)
accuracy_score(y_test,y_predict)
```

/usr/local/lib/python3.10/dist-packages/sklearn/utils/validation.py:1143: 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[14]:

0.9466666666666667

In [15]:

```
print(classification_report(y_test,y_predict))
```

	precision	recall	f1-score	support
0	1.00	1.00	1.00	23
1	0.96	0.89	0.92	27
2	0.89	0.96	0.92	25
accuracy			0.95	75
macro avg	0.95	0.95	0.95	75
weighted avg	0.95	0.95	0.95	75

In [16]:

```
confusion_matrix(y_test,y_predict)
```

Out[16]:

```
array([[23, 0, 0],
       [ 0, 24, 3],
       [ 0, 1, 24]])
```

In [17]:

```
model = SVC(C=0.1)
model.fit(X_train,y_train)
y_predict = model.predict(X_test)
accuracy_score(y_test,y_predict)
```

/usr/local/lib/python3.10/dist-packages/sklearn/utils/validation.py:1143: 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[17]:

0.68

In [18]:

```
model = SVC(C=10)
model.fit(X_train,y_train)
y_predict = model.predict(X_test)
accuracy_score(y_test,y_predict)
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/utils/validation.py:1143: 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[18]:

0.96

In [19]:

```
model = SVC(C=100)
model.fit(X_train,y_train)
y_predict = model.predict(X_test)
accuracy_score(y_test,y_predict)
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/utils/validation.py:1143: 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]:

0.9333333333333333

In [20]:

```
model = SVC(C=1000)
model.fit(X_train,y_train)
y_predict = model.predict(X_test)
accuracy_score(y_test,y_predict)
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/utils/validation.py:1143: 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[20]:

0.9466666666666667

In [21]:

```
model = SVC(gamma=0.01)
model.fit(X_train,y_train)
y_predict = model.predict(X_test)
accuracy_score(y_test,y_predict)
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/utils/validation.py:1143: 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[21]:

0.9066666666666666

In [22]:

```
model = SVC(gamma=0.1)
model.fit(X_train,y_train)
y_predict = model.predict(X_test)
accuracy_score(y_test,y_predict)
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/utils/validation.py:1143: 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().
  v = column_or_1d(v, warn=True)
```

Out[22]:

0.96

In [23]:

```
model = SVC(gamma=1)
model.fit(X_train,y_train)
y_predict = model.predict(X_test)
accuracy_score(y_test,y_predict)
```

/usr/local/lib/python3.10/dist-packages/sklearn/utils/validation.py:1143: 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[23]:

0.9333333333333333

In [24]:

```
model = SVC(gamma=100)
model.fit(X_train,y_train)
y_predict = model.predict(X_test)
accuracy_score(y_test,y_predict)
```

/usr/local/lib/python3.10/dist-packages/sklearn/utils/validation.py:1143: 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[24]:

0.48

In [25]:

```
model = SVC(C=100,gamma=0.1,kernel='poly')
model.fit(X_train,y_train)
y_predict = model.predict(X_test)
accuracy_score(y_test,y_predict)
```

/usr/local/lib/python3.10/dist-packages/sklearn/utils/validation.py:1143: 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[25]:

0.96

In [26]:

```
model = SVC(kernel='linear')
model.fit(X_train,y_train)
y_predict = model.predict(X_test)
accuracy_score(y_test,y_predict)
```

/usr/local/lib/python3.10/dist-packages/sklearn/utils/validation.py:1143: 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[26]:

0.96

In [27]:

```
model = SVC(kernel='poly')
model.fit(X_train,y_train)
y_predict = model.predict(X_test)
```

```
accuracy_score(y_test,y_predict)
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/utils/validation.py:1143: 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[27]:
```

```
0.9466666666666667
```

```
In [28]:
```

```
df0 = data[:50]
df1 = data[50:100]
df2 = data[100:150]
```

```
In [29]:
```

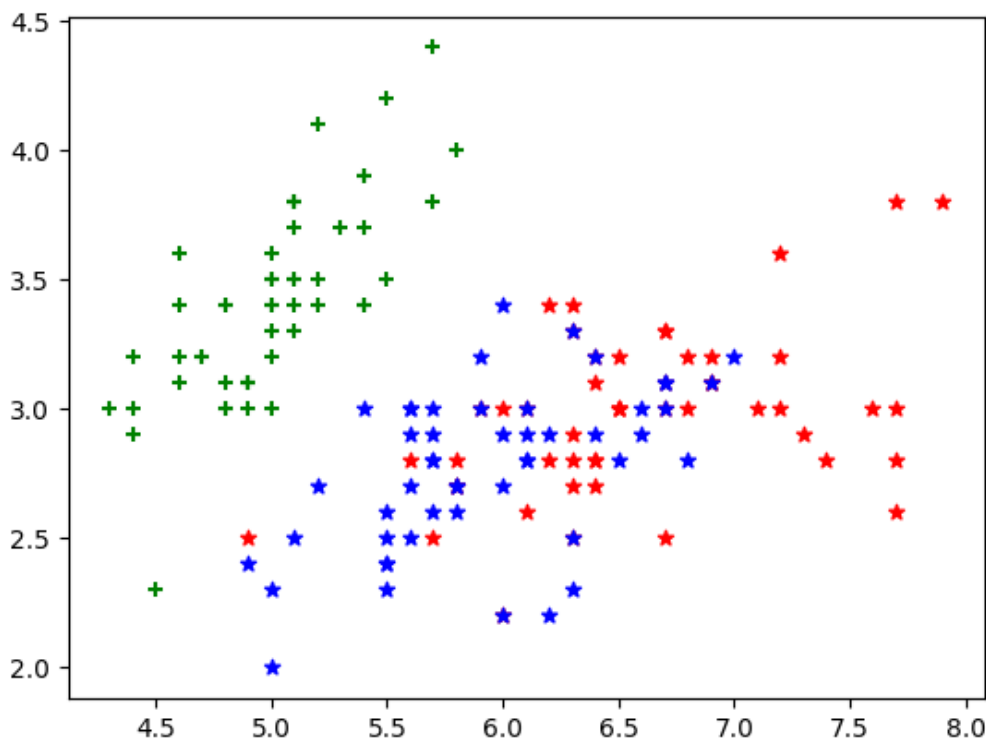
```
x = data.iloc[:100,:2]
y = data.target[:100]
```

```
In [30]:
```

```
plt.scatter(df0['sepal_length'],df0['sepal_width'],color='green',marker='+')
plt.scatter(df2['sepal_length'],df2['sepal_width'],color='red',marker='*')
plt.scatter(df1['sepal_length'],df1['sepal_width'],color='blue',marker='*')
```

```
Out[30]:
```

```
<matplotlib.collections.PathCollection at 0x7af8b7b49630>
```



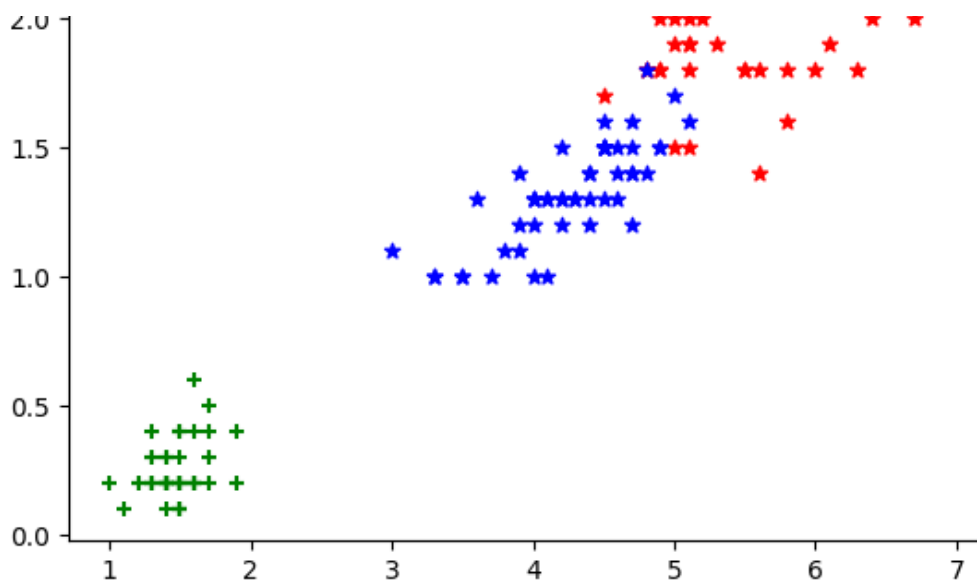
```
In [31]:
```

```
plt.scatter(df0['petal_length'],df0['petal_width'],color='green',marker='+')
plt.scatter(df2['petal_length'],df2['petal_width'],color='red',marker='*')
plt.scatter(df1['petal_length'],df1['petal_width'],color='blue',marker='*')
```

```
Out[31]:
```

```
<matplotlib.collections.PathCollection at 0x7af8b5a84850>
```



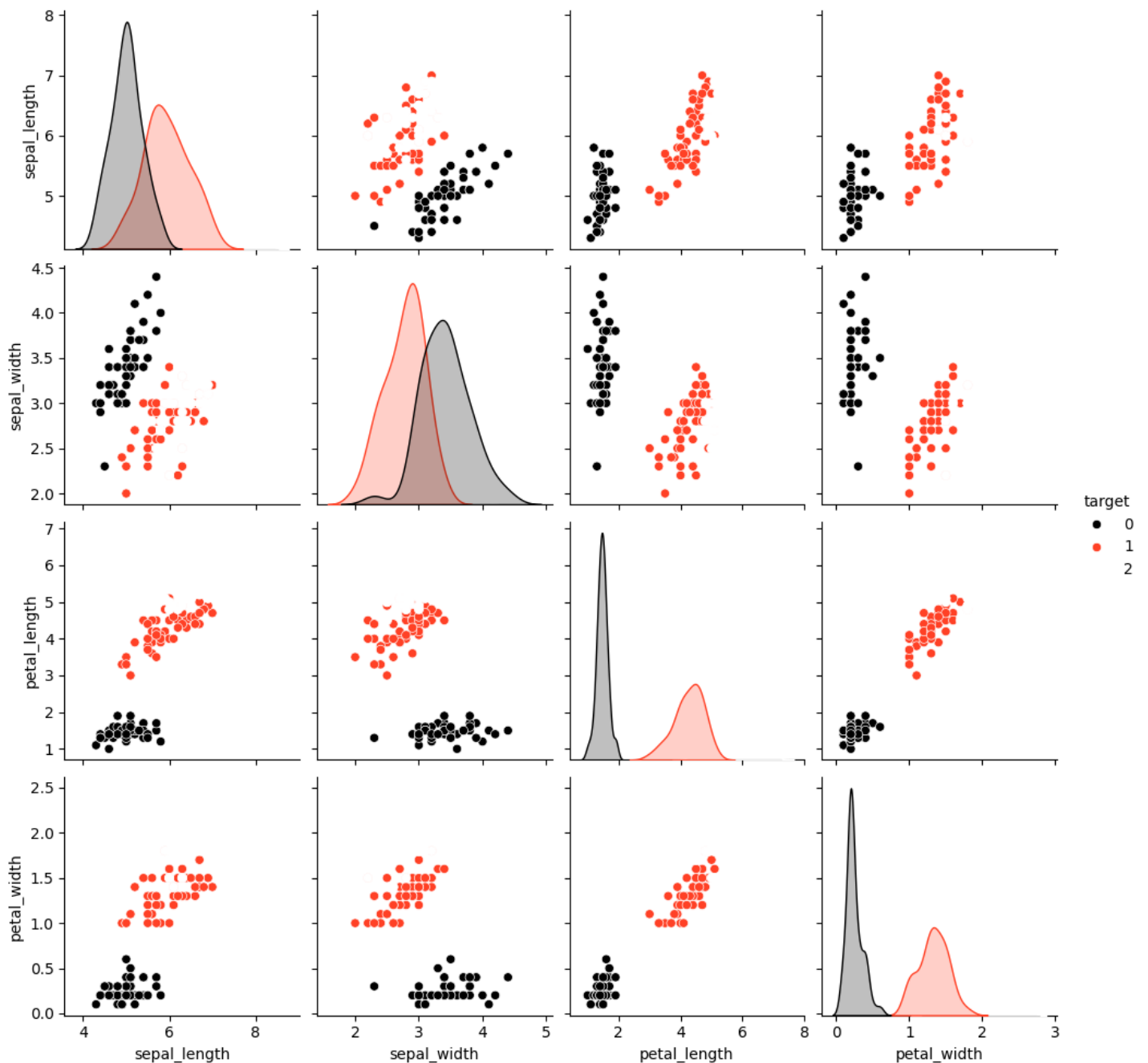


In [32]:

```
import seaborn as sns
sns.pairplot(data, hue="target", palette="CMRmap")
```

Out[32]:

<seaborn.axisgrid.PairGrid at 0x7af8b5aa7d90>



In [33]:

```
import seaborn as sns
sns.pairplot(data, hue="target", palette="rainbow")
```

Out[33]:

<seaborn.axisgrid.PairGrid at 0x7af8b2125f90>

