In [5]:

```
import numpy as np
import pandas as pd
```

Iris Dataset

In [6]:

```
df = pd.read_csv('Iris.csv')
```

In [7]:

```
df.head()
```

Out[7]:

	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	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 [8]:

```
df.describe()
```

Out[8]:

	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm
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 [9]:

```
X = df.drop(columns='Species' , axis = 1)
```

In [10]:

```
from sklearn.cluster import KMeans
km = KMeans(n_clusters = 3, random_state = 1)
km.fit(X)
```

```
C:\Users\Prajwal\AppData\Roaming\Python\Python311\site-packages\sklearn\cluster\_kme
ans.py:1416: FutureWarning: The default value of `n_init` will change from 10 to 'au
to' in 1.4. Set the value of `n_init` explicitly to suppress the warning
  super()._check_params_vs_input(X, default_n_init=10)
C:\Users\Prajwal\AppData\Roaming\Python\Python311\site-packages\joblib\externals\lok
y\backend\context.py:136: UserWarning: Could not find the number of physical cores f
or the following reason:
[WinError 2] The system cannot find the file specified
Returning the number of logical cores instead. You can silence this warning by setti
ng LOKY_MAX_CPU_COUNT to the number of cores you want to use.
  warnings.warn(
  File "C:\Users\Prajwal\AppData\Roaming\Python\Python311\site-packages\joblib\exter
nals\loky\backend\context.py", line 257, in _count_physical_cores
    cpu_info = subprocess.run(
               ^^^^^^^^^^^^^^^
  File "C:\Users\Prajwal\AppData\Local\Programs\Python\Python311\Lib\subprocess.py",
line 546, in run
    with Popen(*popenargs, **kwargs) as process:
         ^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "C:\Users\Prajwal\AppData\Local\Programs\Python\Python311\Lib\subprocess.py",
line 1022, in __init__
    self._execute_child(args, executable, preexec_fn, close_fds,
  File "C:\Users\Prajwal\AppData\Local\Programs\Python\Python311\Lib\subprocess.py",
line 1491, in _execute_child
    hp, ht, pid, tid = _winapi.CreateProcess(executable, args,
                       ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
```

Out[10]:

```
▾ KMeans
KMeans(n_clusters=3, random_state=1)
```

In [11]:

```python
km.labels_
```

Out[11]:

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

In [12]:

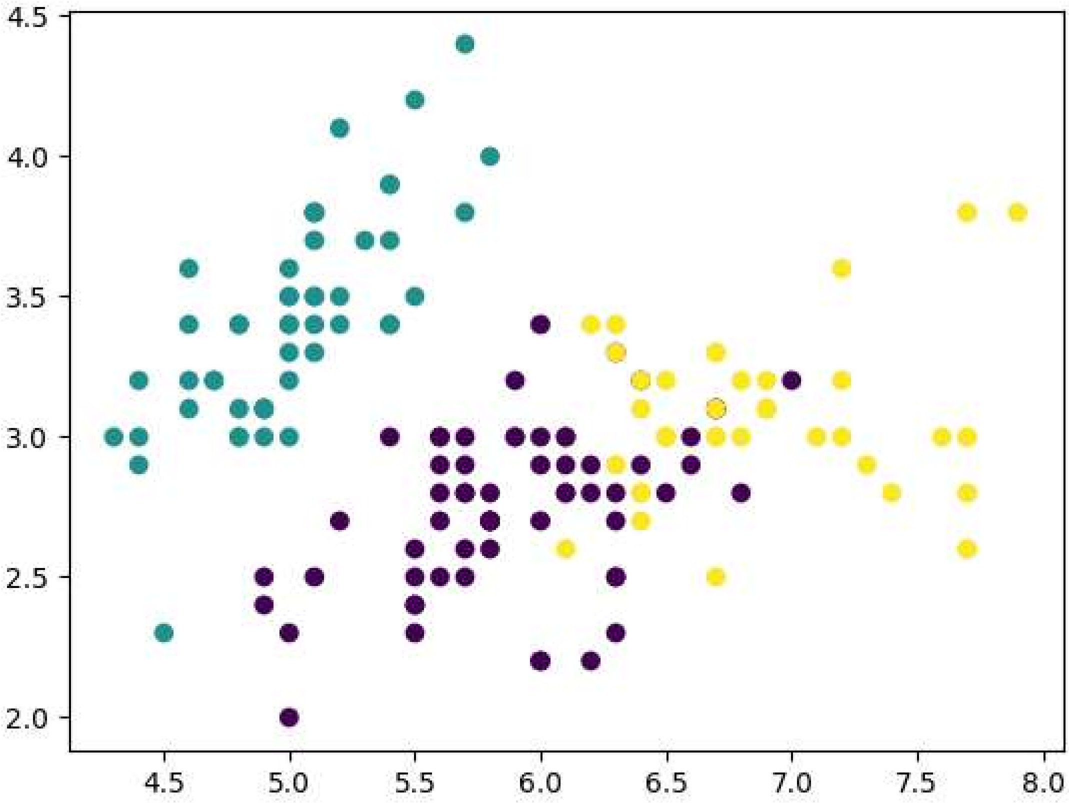
```python
km.cluster_centers_
```

Out[12]:

```
array([[5.9016129 , 2.7483871 , 4.39354839, 1.43387097],
       [5.006     , 3.418     , 1.464     , 0.244     ],
       [6.85      , 3.07368421, 5.74210526, 2.07105263]])
```

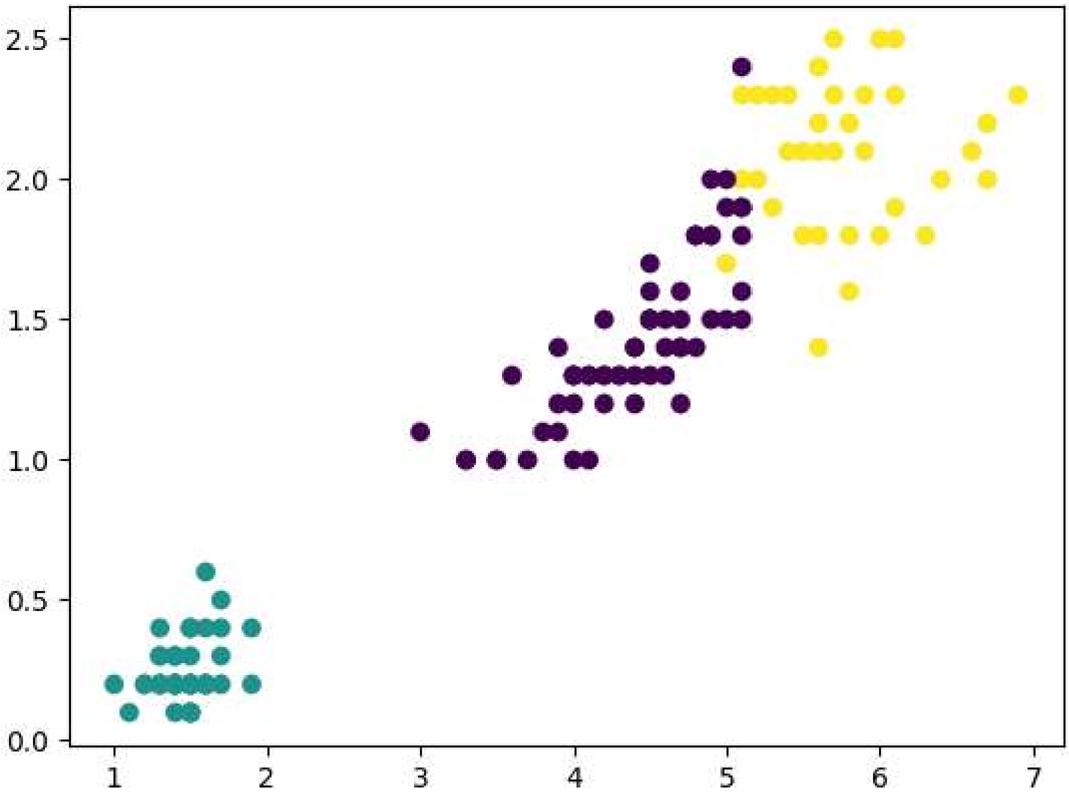
In [14]:

```python
import matplotlib.pyplot as plt
plt.scatter(X['SepalLengthCm'], X['SepalWidthCm'], c = km.labels_)
plt.show()
```



In [15]:

```
plt.scatter(X['PetalLengthCm'], X['PetalWidthCm'], c = km.labels_)
plt.show()
```



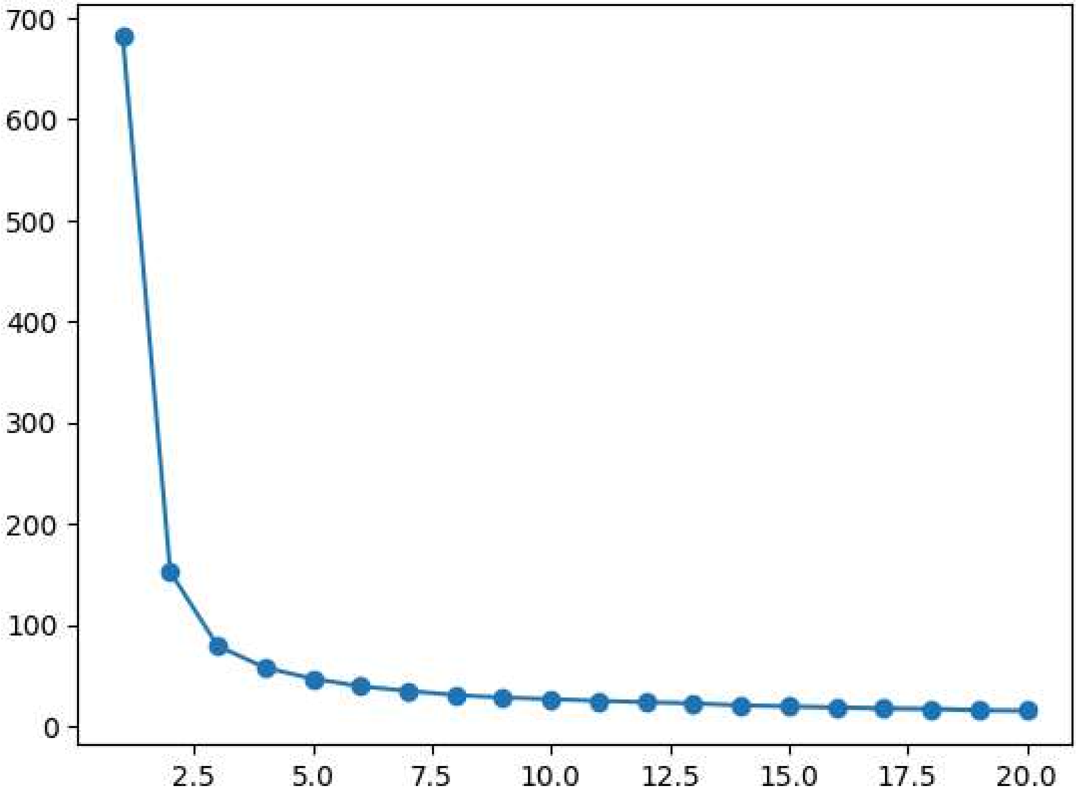
In [ ]:

```
inertia_scores = []
for i in range(1,21):
    k = KMeans(n_clusters = i)
    k.fit(X)
    inertia_scores.append(k.inertia_)
```

```
In [17]: inertia_scores
```

```
Out[17]: [680.8244,
          152.36870647733906,
          78.94084142614601,
          57.34540931571816,
          46.535582051282056,
          38.930963049671746,
          34.382179288987984,
          30.200787597335584,
          27.92560087719298,
          26.15720238095238,
          24.58495644331514,
          23.142619658119656,
          22.080556524393483,
          20.07232440353764,
          19.188786899375135,
          18.13599206349207,
          17.23896816908582,
          16.794532245532245,
          15.526906141244378,
          14.996695324283563]
```

```
In [21]: x_labels = range(1,21)
         plt.plot(x_labels,inertia_scores,marker='o')
         plt.show()
```



```
In [22]: from sklearn.cluster import AgglomerativeClustering
```

```
In [23]: cluster = AgglomerativeClustering(n_clusters = 3).fit(X)
         cluster
```

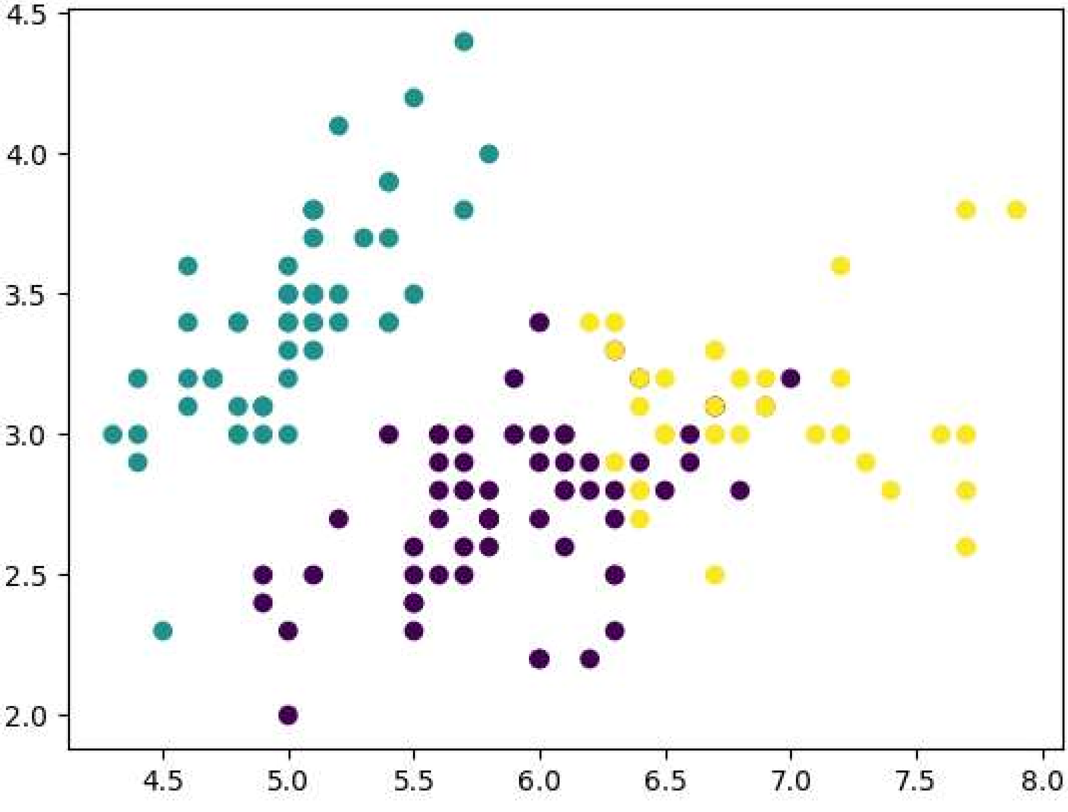
```
Out[23]: ▾        AgglomerativeClustering

         AgglomerativeClustering(n_clusters=3)
```

```
In [24]: cluster.labels_
```

```
Out[24]: array([1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
                1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
                1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
                0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
                0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 2, 0, 2, 2, 2, 2, 0, 2, 2, 2,
                2, 2, 2, 0, 0, 2, 2, 2, 2, 0, 2, 0, 2, 0, 2, 2, 0, 0, 2, 2, 2, 2,
                2, 0, 0, 2, 2, 2, 0, 2, 2, 2, 0, 2, 2, 2, 0, 2, 2, 0], dtype=int64)
```

```
In [25]: plt.scatter(X['SepalLengthCm'], X['SepalWidthCm'], c = cluster.labels_)
         plt.show()
```



```
In [28]: from sklearn.metrics import confusion_matrix
         confusion_matrix(km.labels_,cluster.labels_)
```

```
Out[28]: array([[62,  0,  0],
                [ 0, 50,  0],
                [ 2,  0, 36]], dtype=int64)
```

In [29]:

```python
df = pd.DataFrame({'km' : km.labels_, 'ac' : cluster.labels_})
df
```

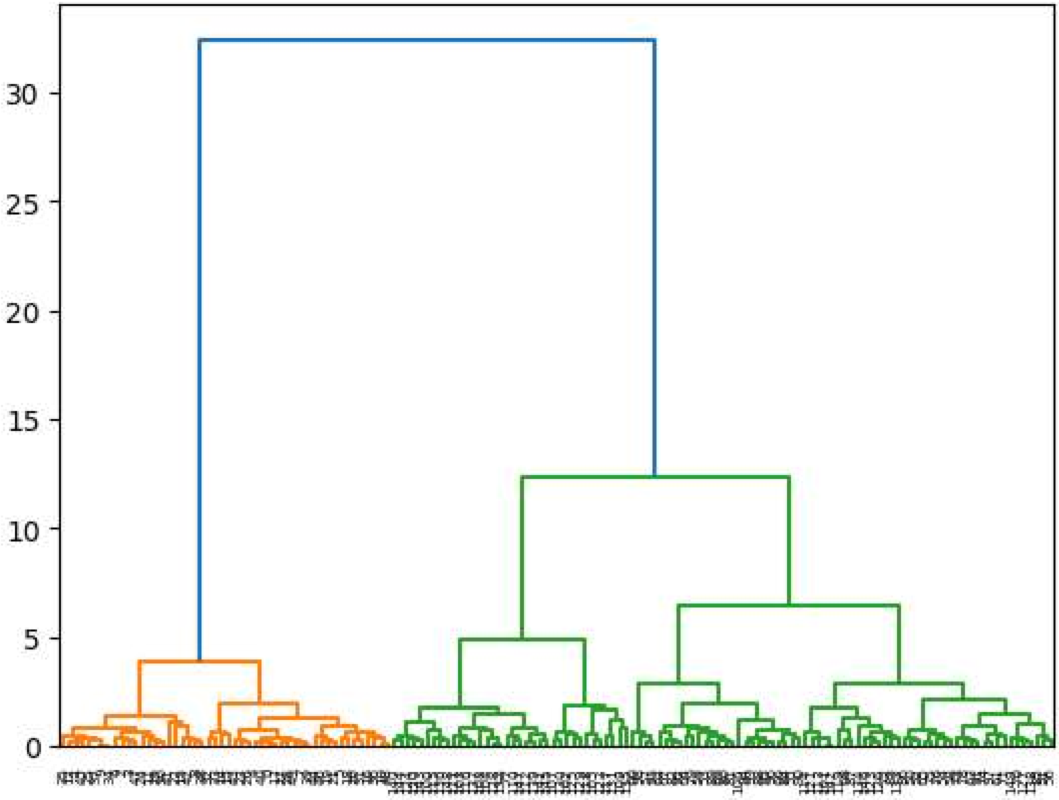
Out[29]:

	km	ac
0	1	1
1	1	1
2	1	1
3	1	1
4	1	1
...	...	...
145	2	2
146	0	0
147	2	2
148	2	2
149	0	0

150 rows × 2 columns

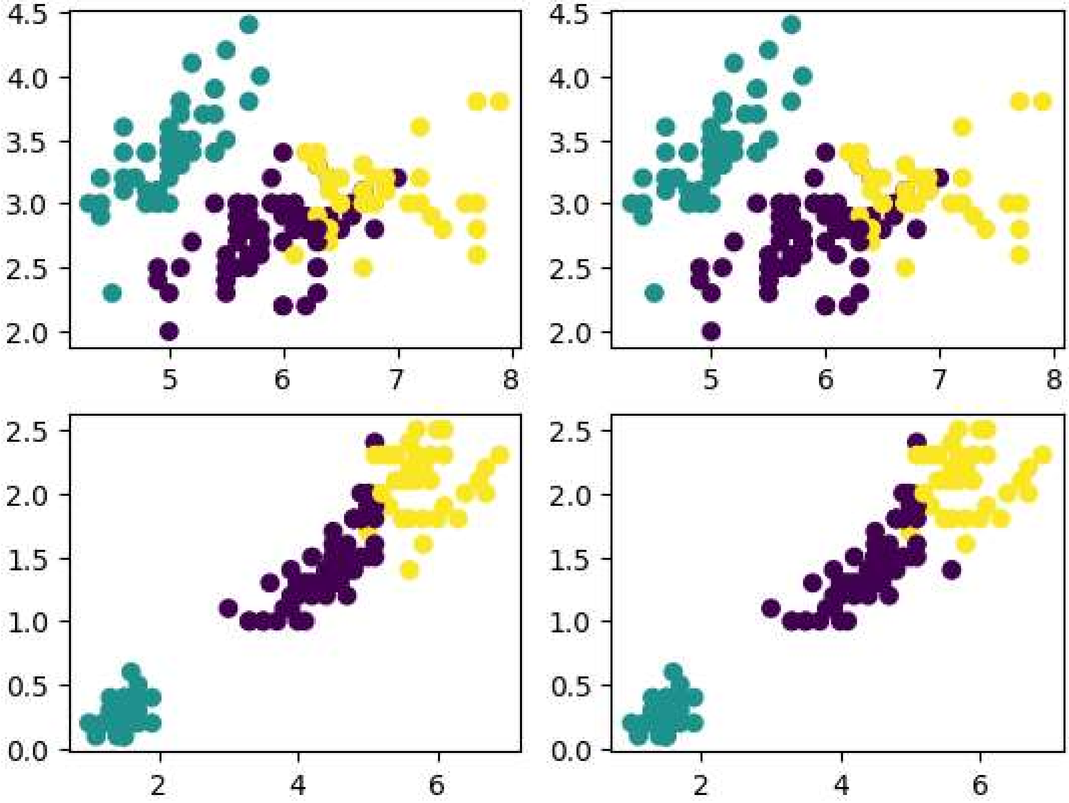
In [31]:

```python
import scipy.cluster.hierarchy as sch
dendrogram = sch.dendrogram(sch.linkage(X, method = 'ward'))
```



```
In [33]: plt.subplot(2, 2, 1)
         plt.scatter(X['SepalLengthCm'], X['SepalWidthCm'], c = km.labels_)
         plt.subplot(2, 2, 2)
         plt.scatter(X['SepalLengthCm'], X['SepalWidthCm'], c = cluster.labels_)
         plt.subplot(2, 2, 3)
         plt.scatter(X['PetalLengthCm'], X['PetalWidthCm'], c = km.labels_)
         plt.subplot(2, 2, 4)
         plt.scatter(X['PetalLengthCm'], X['PetalWidthCm'], c = cluster.labels_)
```

```
Out[33]: <matplotlib.collections.PathCollection at 0x26c76c8fc90>
```


4.5
4.0
3.5
3.0
2.5
2.0
5
6
7
8
4.5
4.0
3.5
3.0
2.5
2.0
5
6
7
8
2.5
2.0
1.5
1.0
0.5
0.0
2
4
6
2.5
2.0
1.5
1.0
0.5
0.0
2
4
6