In [1]:

```
from math import sqrt
from random import randint
import numpy as np

class KMC:
    def __init__(self,n_clusters=3,max_iter=5):
        self.n_clusters = n_clusters
        self.max_iter = max_iter
        self.centroids = []
        self.clusters = dict()
        self.labels = []

    def initialize_centroids(self,data):
        T = []
        p = 0
        size = len(data)
        self.labels = [-1]*size
        while p < self.n_clusters:
            f = randint(0,size-1)
            T.append(data[f])
            p += 1
        self.centroids = T

    def get_closest(self,data):
        length = len(data)
        L = []
        for i in range(self.n_clusters):
            dist = 0
            for j in range(length):
                dist += (data[j] - self.centroids[i][j])**2
            dist = sqrt(dist)
            L.append(dist)
        return L.index(min(L))

    def get_centroids(self):
        self.centroids = []
        for cluster in self.clusters.keys():
            T = []
            for i in range(len(self.clusters[cluster][0])):
                s = 0.0
                for j in range(len(self.clusters[cluster])):
                    s += self.clusters[cluster][j][i]
                T.append(s/len(self.clusters[cluster]))
            self.centroids.append(T)

    def fit(self,data):
        self.initialize_centroids(data)

        for i in range(len(data)):
            x = self.get_closest(data[i])
            self.labels[i] = x
            if x not in self.clusters.keys():
                self.clusters[x] = []
            self.clusters[x].append(data[i])
```

```
    self.get_centroids()

    for i in range(20):
        d = dict()
        for j in range(len(data)):
            y = self.get_closest(data[j])
            self.labels[j] = y
            if y not in d.keys():
                d[y] = []
            d[y].append(data[j])
        self.clusters = dict()
        self.clusters = d
        self.get_centroids()
        if i == 4 or i == 9 or i == 19:
            print str(self.n_clusters) + ' ' + str(i) + ' ' + str(self.get_ss
    self.labels = np.asarray(self.labels)

def get_distance(self,centroid,point):
    x = len(centroid)
    w = 0.0
    for i in range(x):
        w += (centroid[i] - point[i])**2
    return w

def get_sstotal(self):
    s = 0.0
    for cluster in self.clusters.keys():
        for point in self.clusters[cluster]:
            s += self.get_distance(self.centroids[cluster],point)
    return s
```

In [2]:

```
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D
from sklearn import datasets
import numpy as np

iris = datasets.load_iris()
X = iris.data
Y = iris.target

K = KMC()
K.fit(X)
```

```
3 4 78.9408414261
3 9 78.9408414261
3 19 78.9408414261
```

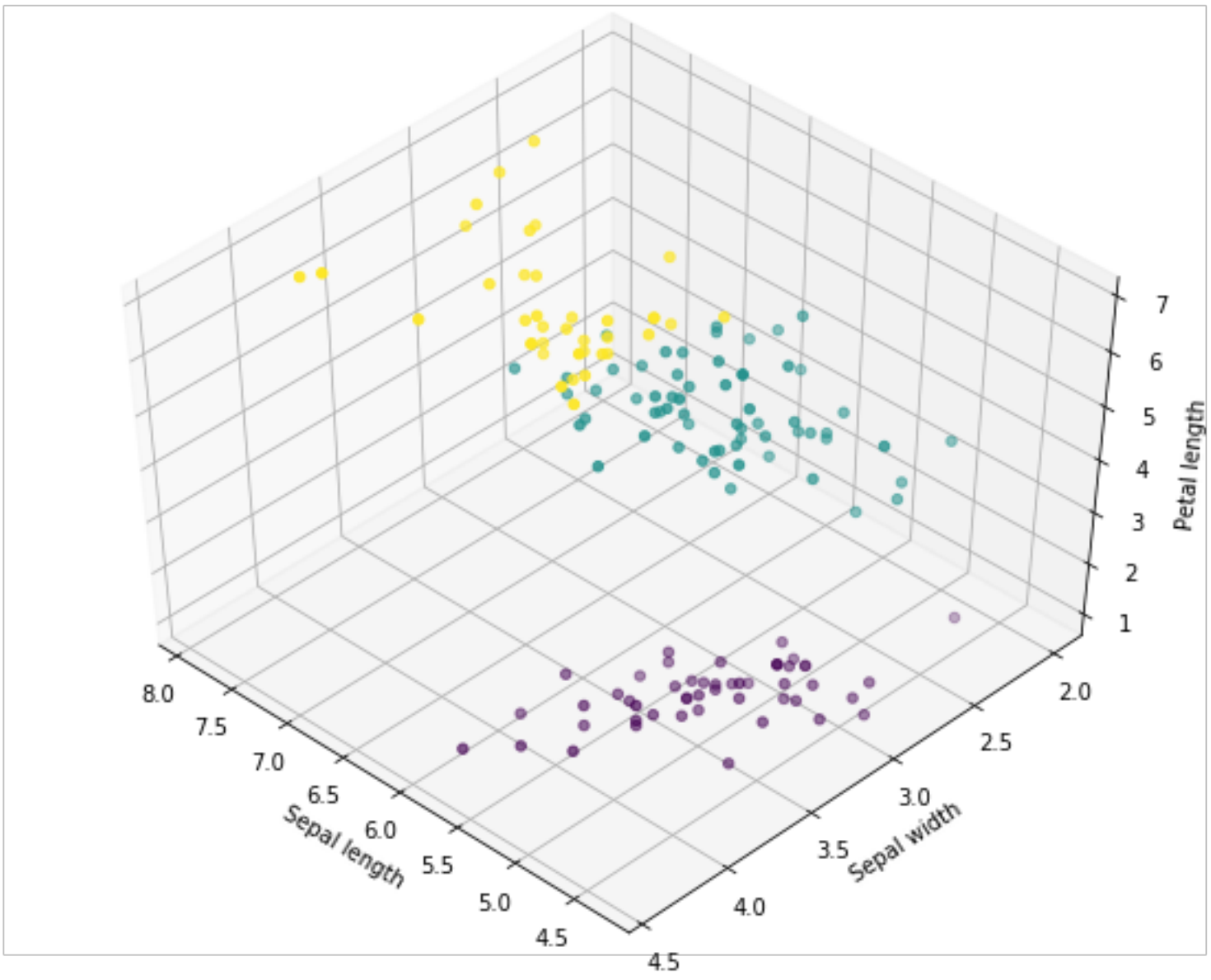
In [3]:

```
fig = plt.figure(1,figsize=(8,6))
plt.clf()
ax = Axes3D(fig,rect=[0,0,0.95,1],elev=48,azim=134)

plt.cla()
ax.scatter(X[:,[0]],X[:,[1]],X[:,[2]],c=K.labels.astype(np.float))

ax.set_xlabel('Sepal length')
ax.set_ylabel('Sepal width')
ax.set_zlabel('Petal length')

plt.show()
```



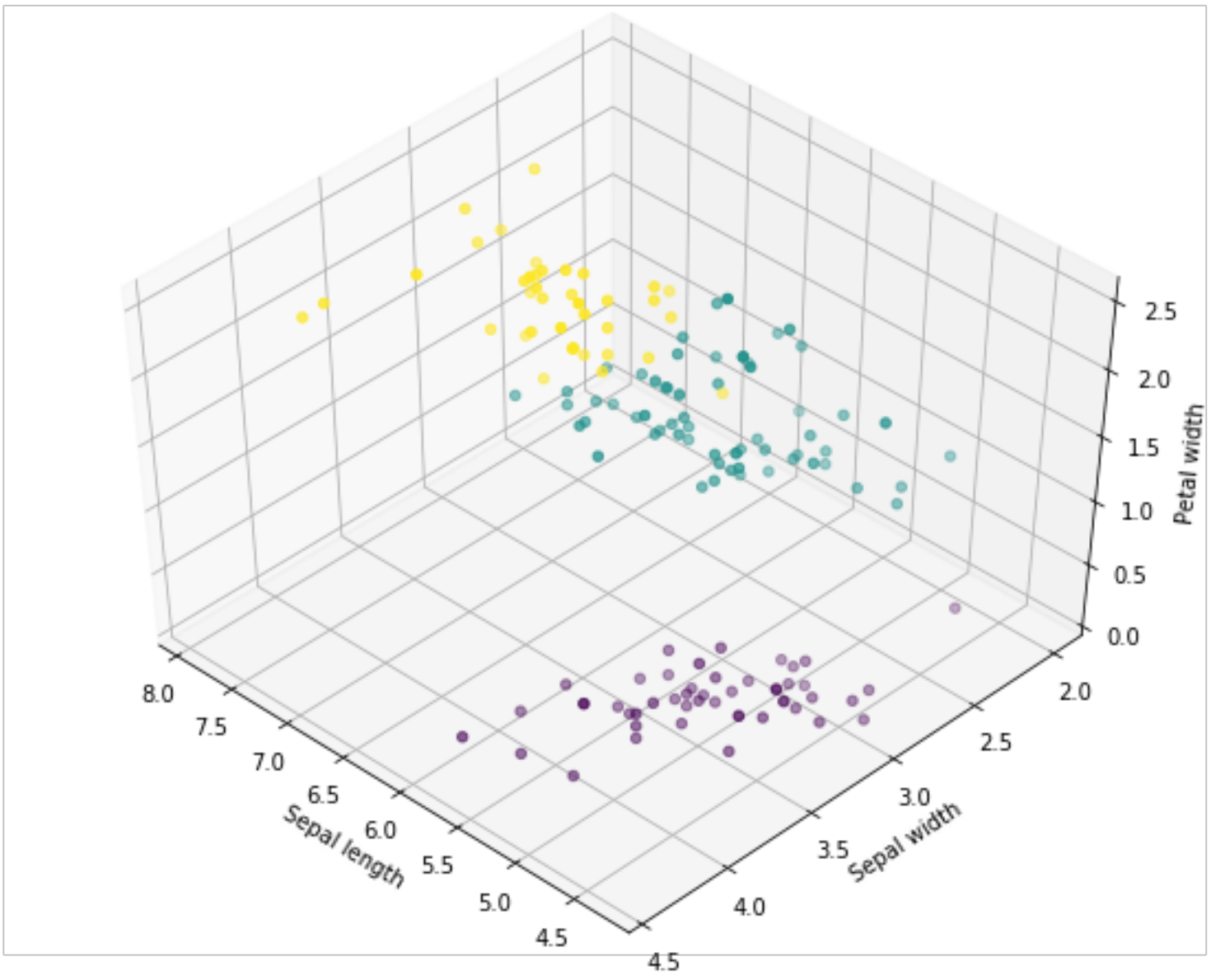
In [4]:

```
fig = plt.figure(1,figsize=(8,6))
plt.clf()
ax = Axes3D(fig,rect=[0,0,0.95,1],elev=48,azim=134)

plt.cla()
ax.scatter(X[:,[0]],X[:,[1]],X[:,[3]],c=K.labels.astype(np.float))

ax.set_xlabel('Sepal length')
ax.set_ylabel('Sepal width')
ax.set_zlabel('Petal width')

plt.show()
```



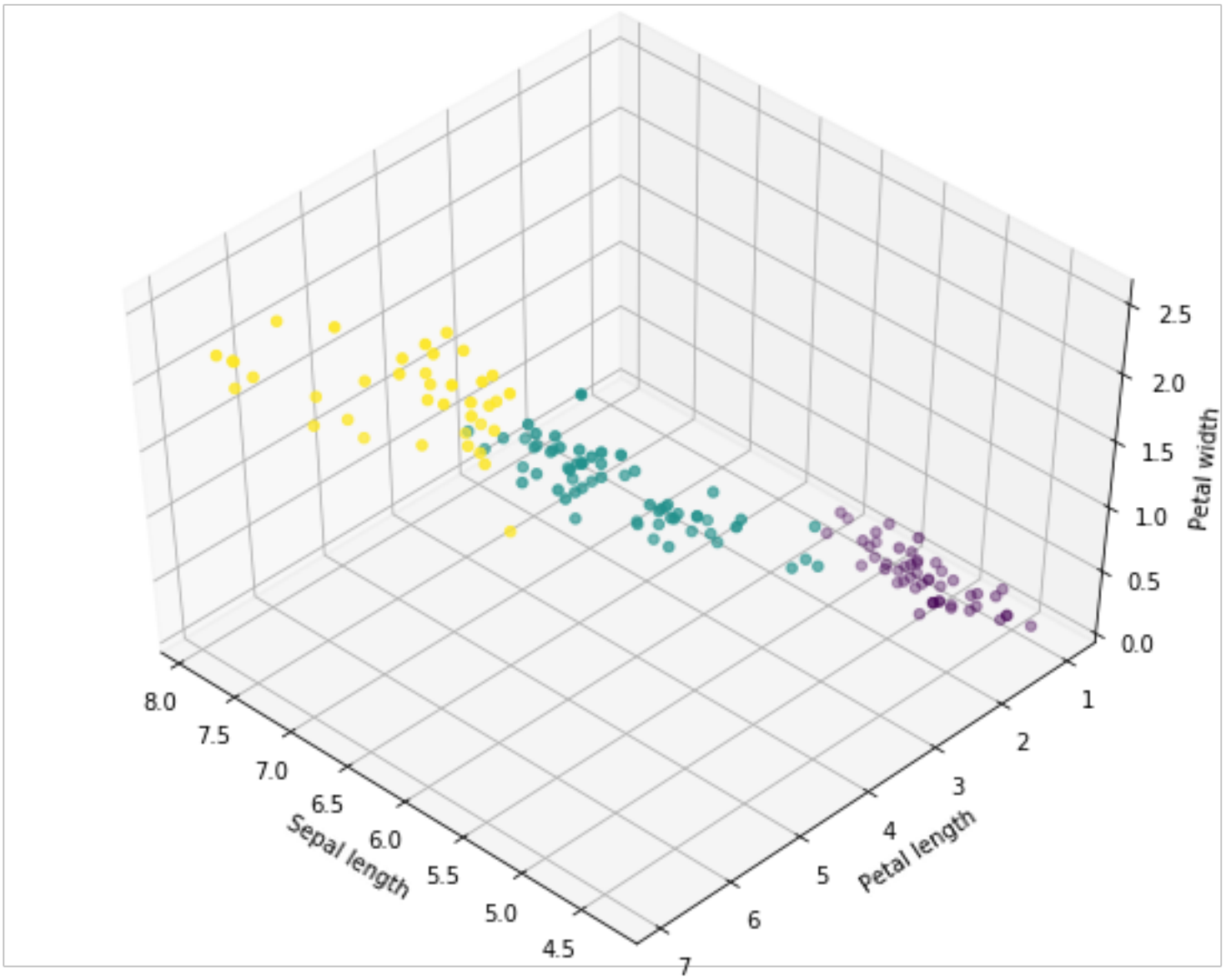
In [5]:

```
fig = plt.figure(1,figsize=(8,6))
plt.clf()
ax = Axes3D(fig,rect=[0,0,0.95,1],elev=48,azim=134)

plt.cla()
ax.scatter(X[:,[0]],X[:,[2]],X[:,[3]],c=K.labels.astype(np.float))

ax.set_xlabel('Sepal length')
ax.set_ylabel('Petal length')
ax.set_zlabel('Petal width')

plt.show()
```



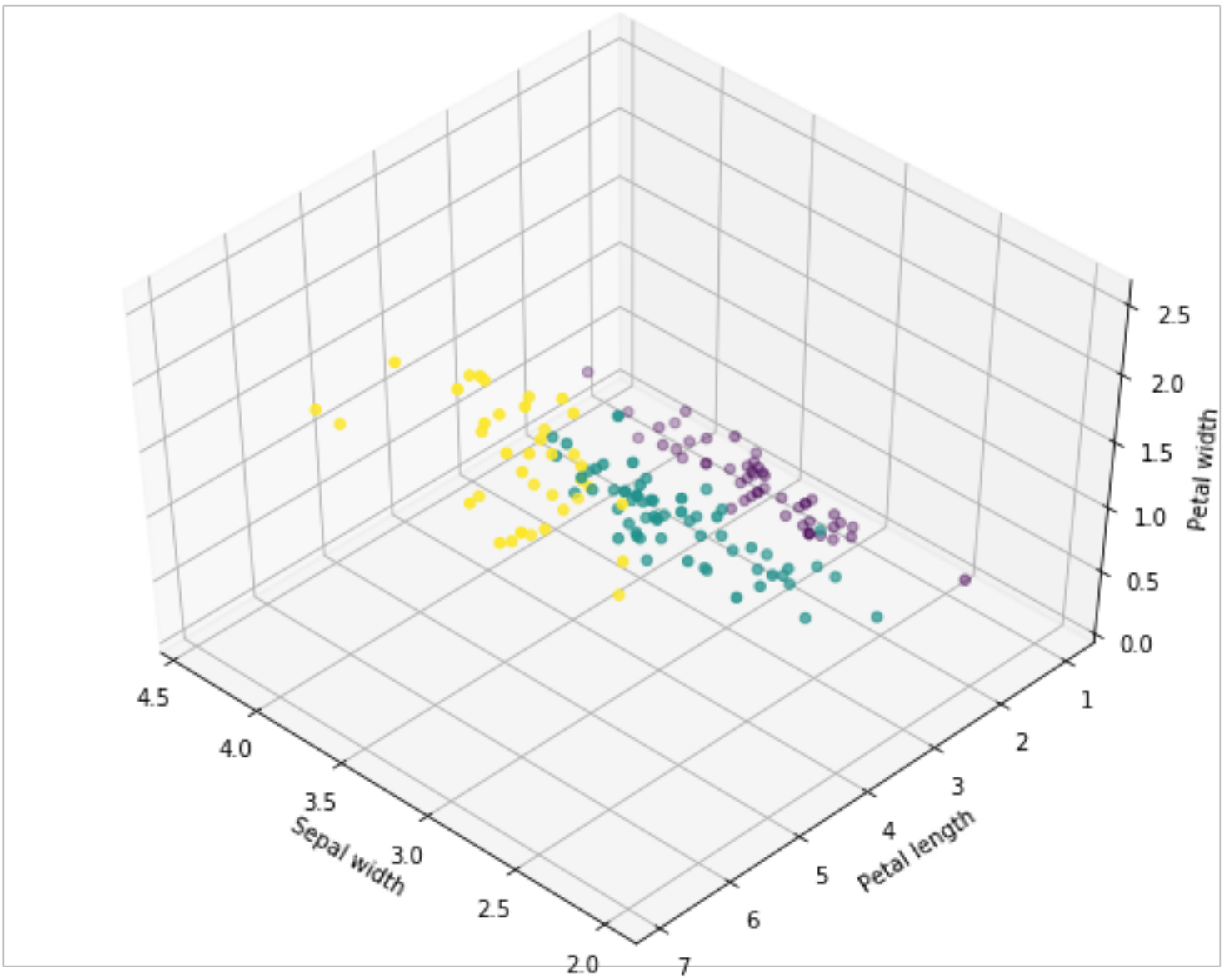
In [6]:

```
fig = plt.figure(1,figsize=(8,6))
plt.clf()
ax = Axes3D(fig,rect=[0,0,0.95,1],elev=48,azim=134)

plt.cla()
ax.scatter(X[:,[1]],X[:,[2]],X[:,[3]],c=K.labels.astype(np.float))

ax.set_xlabel('Sepal width')
ax.set_ylabel('Petal length')
ax.set_zlabel('Petal width')

plt.show()
```



In [7]:

```
M = KMC(5)
M.fit(X)
```

```
5 4 50.3190709677
5 9 49.95165
5 19 49.95165
```

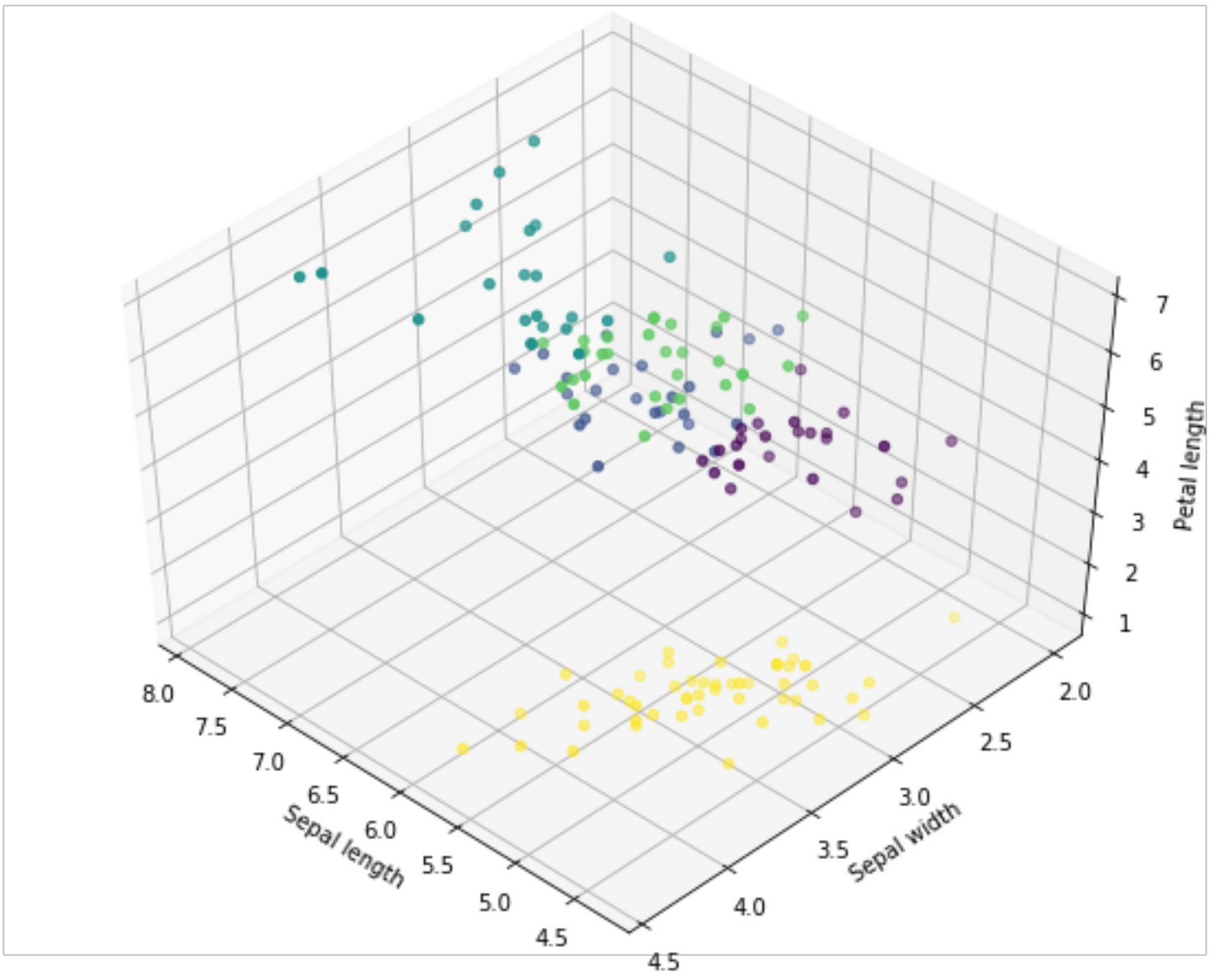
In [8]:

```
fig = plt.figure(1,figsize=(8,6))
plt.clf()
ax = Axes3D(fig,rect=[0,0,0.95,1],elev=48,azim=134)

plt.cla()
ax.scatter(X[:,[0]],X[:,[1]],X[:,[2]],c=M.labels.astype(np.float))

ax.set_xlabel('Sepal length')
ax.set_ylabel('Sepal width')
ax.set_zlabel('Petal length')

plt.show()
```



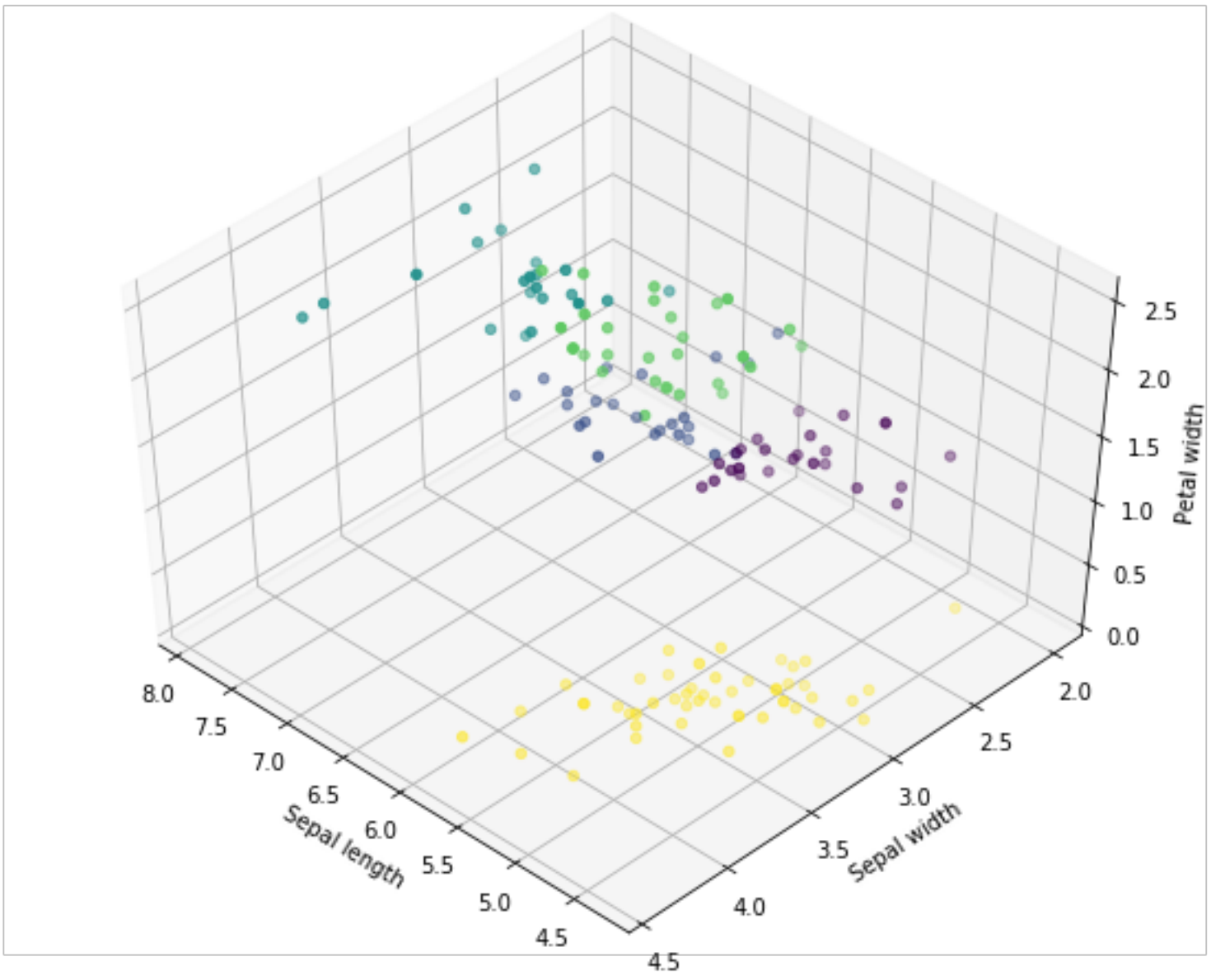
In [9]:

```
fig = plt.figure(1,figsize=(8,6))
plt.clf()
ax = Axes3D(fig,rect=[0,0,0.95,1],elev=48,azim=134)

plt.cla()
ax.scatter(X[:,[0]],X[:,[1]],X[:,[3]],c=M.labels.astype(np.float))

ax.set_xlabel('Sepal length')
ax.set_ylabel('Sepal width')
ax.set_zlabel('Petal width')

plt.show()
```



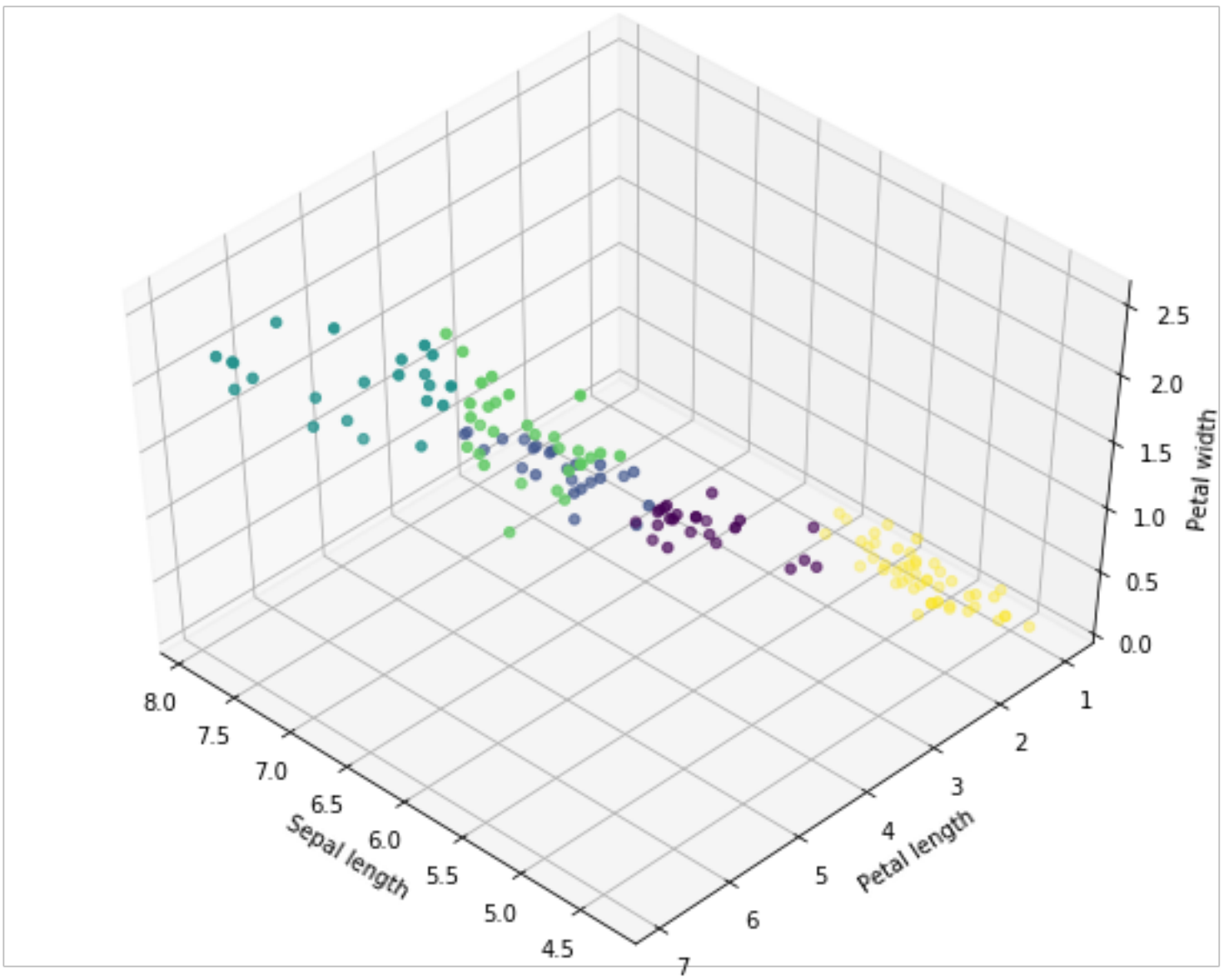
In [10]:

```
fig = plt.figure(1,figsize=(8,6))
plt.clf()
ax = Axes3D(fig,rect=[0,0,0.95,1],elev=48,azim=134)

plt.cla()
ax.scatter(X[:,[0]],X[:,[2]],X[:,[3]],c=M.labels.astype(np.float))

ax.set_xlabel('Sepal length')
ax.set_ylabel('Petal length')
ax.set_zlabel('Petal width')

plt.show()
```



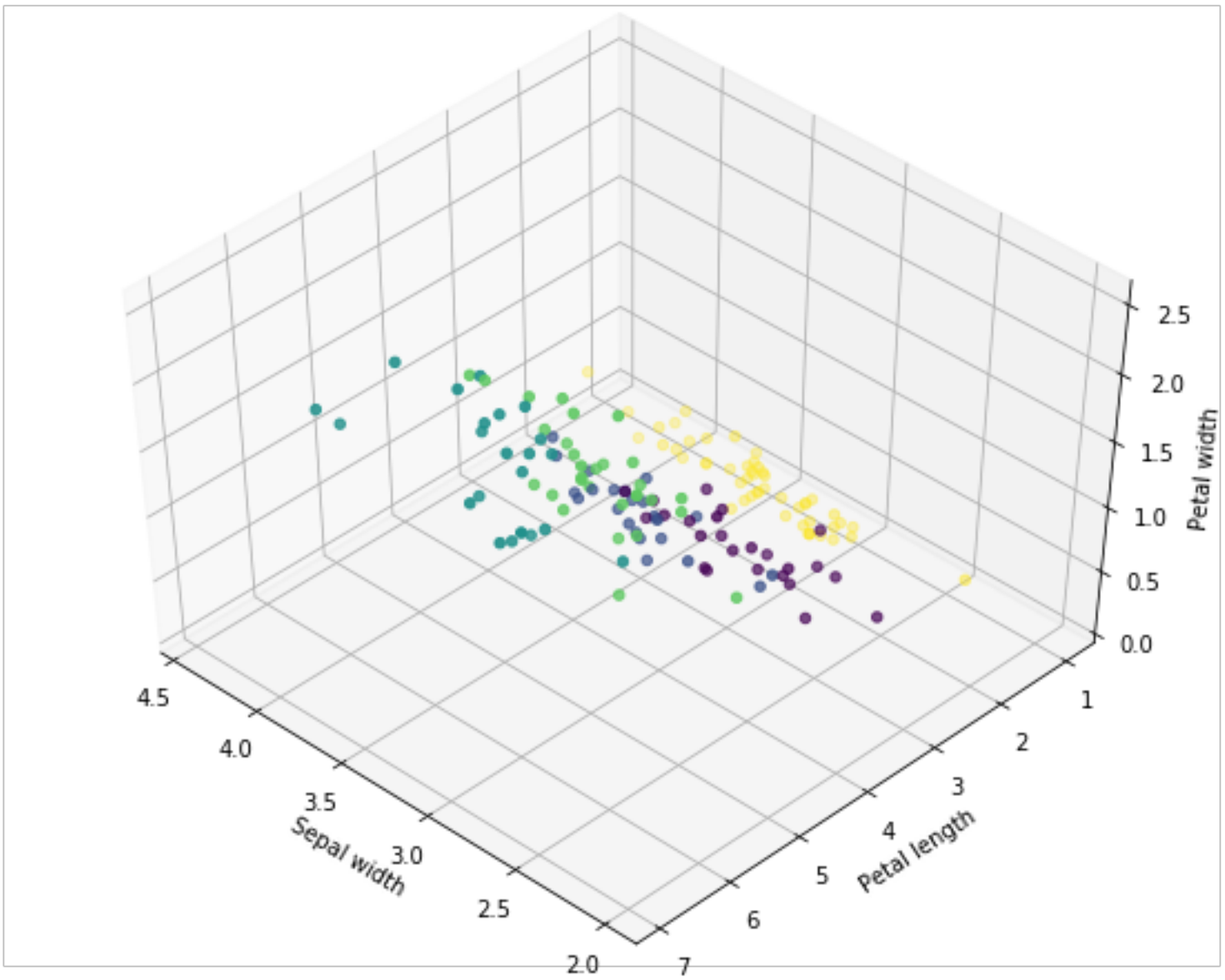
In [11]:

```
fig = plt.figure(1,figsize=(8,6))
plt.clf()
ax = Axes3D(fig,rect=[0,0,0.95,1],elev=48,azim=134)

plt.cla()
ax.scatter(X[:,[1]],X[:,[2]],X[:,[3]],c=M.labels.astype(np.float))

ax.set_xlabel('Sepal width')
ax.set_ylabel('Petal length')
ax.set_zlabel('Petal width')

plt.show()
```



In [12]:

```
W = KMC(7)
W.fit(X)
```

```
7 4 34.4654090177
7 9 34.4654090177
7 19 34.4654090177
```

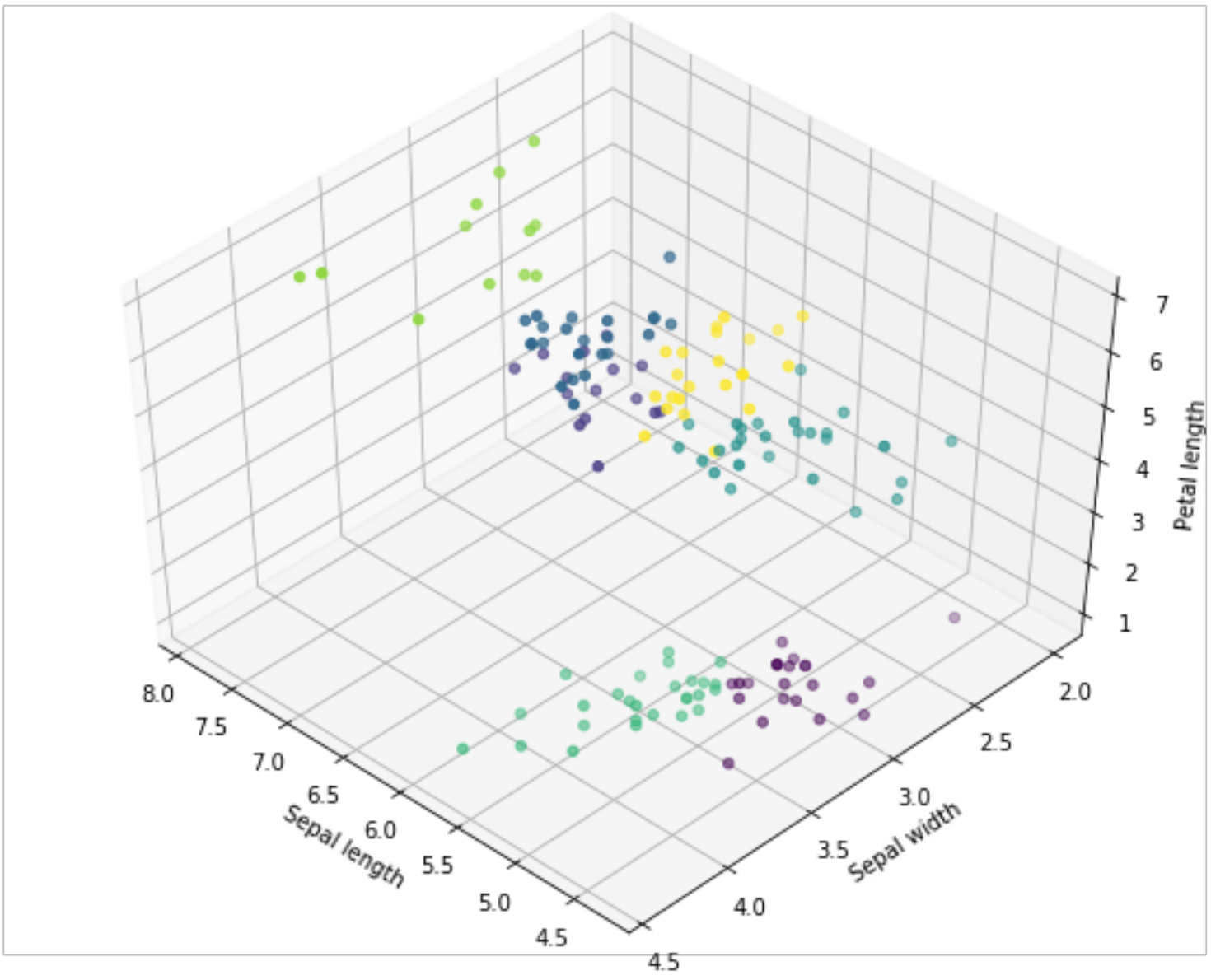
In [13]:

```
fig = plt.figure(1,figsize=(8,6))
plt.clf()
ax = Axes3D(fig,rect=[0,0,0.95,1],elev=48,azim=134)

plt.cla()
ax.scatter(X[:,[0]],X[:,[1]],X[:,[2]],c=W.labels.astype(np.float))

ax.set_xlabel('Sepal length')
ax.set_ylabel('Sepal width')
ax.set_zlabel('Petal length')

plt.show()
```



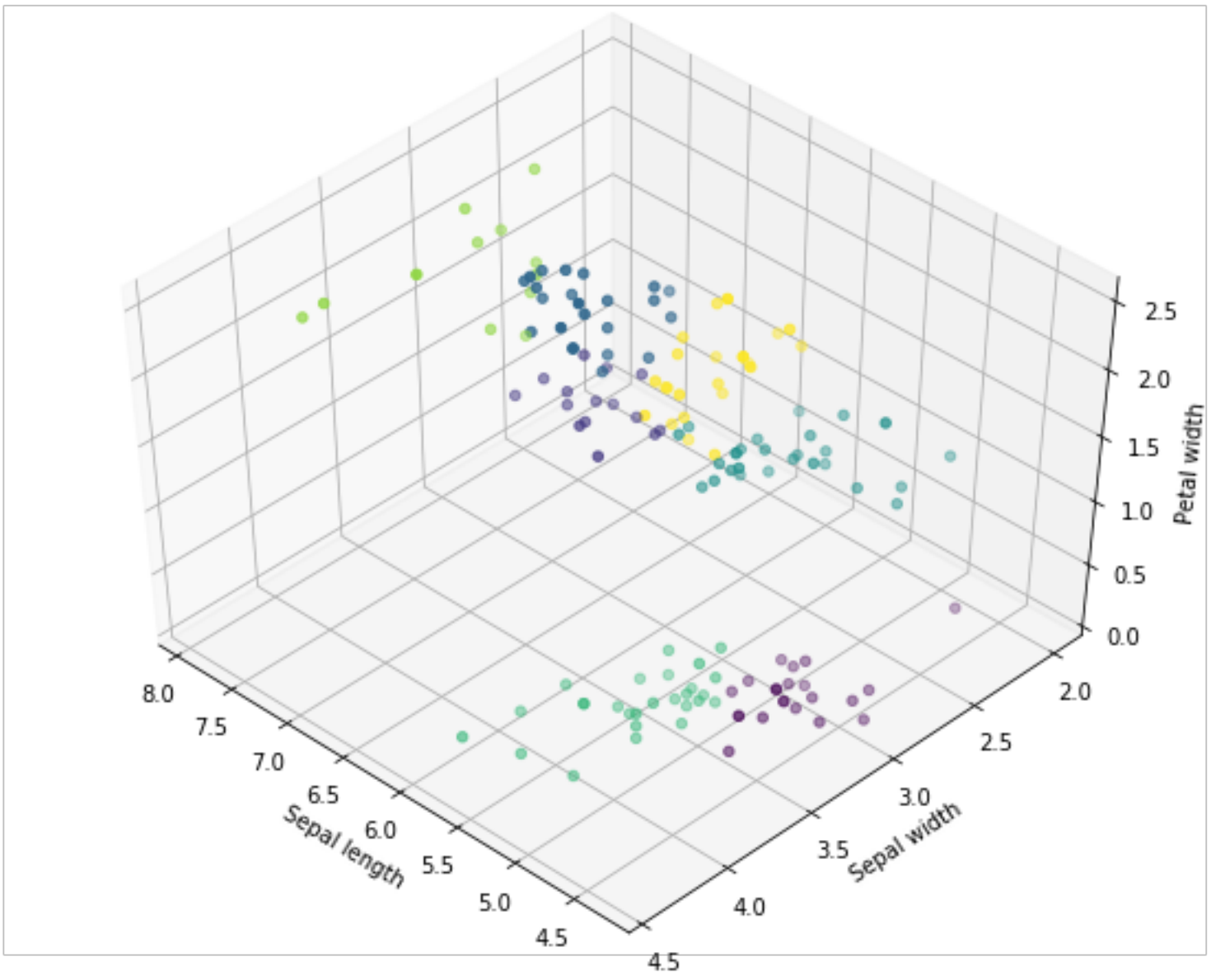
In [14]:

```
fig = plt.figure(1,figsize=(8,6))
plt.clf()
ax = Axes3D(fig,rect=[0,0,0.95,1],elev=48,azim=134)

plt.cla()
ax.scatter(X[:,[0]],X[:,[1]],X[:,[3]],c=W.labels.astype(np.float))

ax.set_xlabel('Sepal length')
ax.set_ylabel('Sepal width')
ax.set_zlabel('Petal width')

plt.show()
```



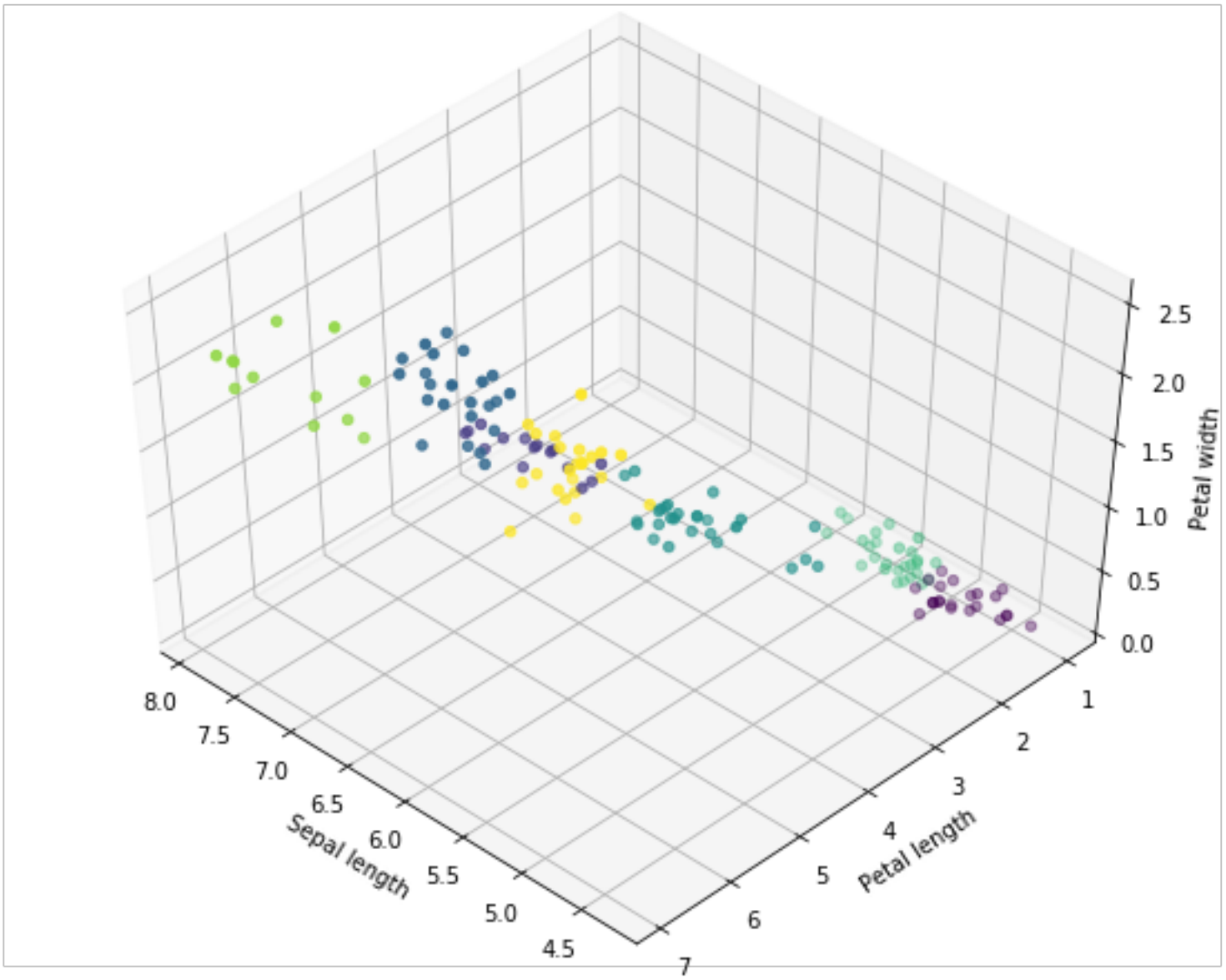
In [15]:

```
fig = plt.figure(1,figsize=(8,6))
plt.clf()
ax = Axes3D(fig,rect=[0,0,0.95,1],elev=48,azim=134)

plt.cla()
ax.scatter(X[:,[0]],X[:,[2]],X[:,[3]],c=W.labels.astype(np.float))

ax.set_xlabel('Sepal length')
ax.set_ylabel('Petal length')
ax.set_zlabel('Petal width')

plt.show()
```



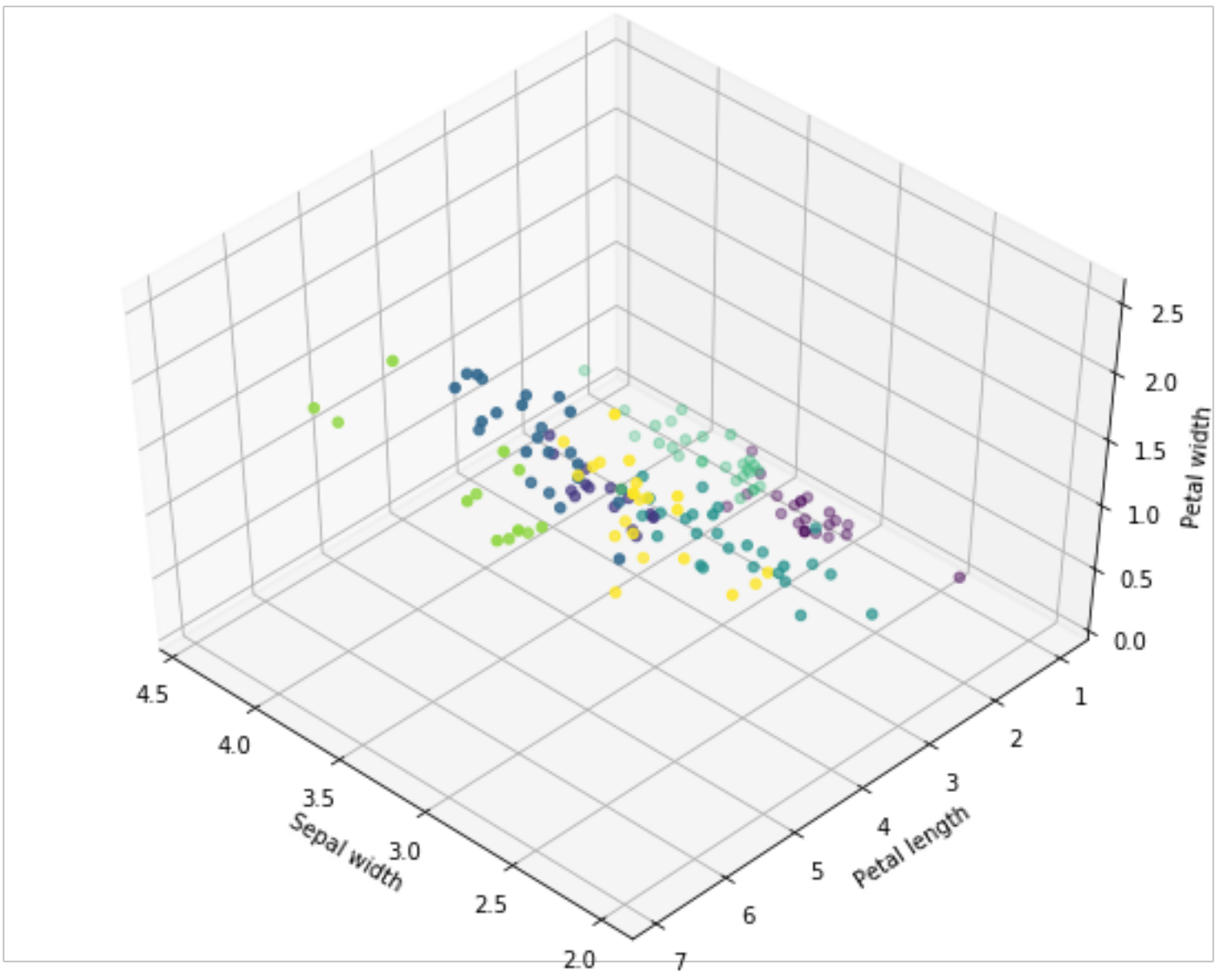
In [16]:

```
fig = plt.figure(1,figsize=(8,6))
plt.clf()
ax = Axes3D(fig,rect=[0,0,0.95,1],elev=48,azim=134)

plt.cla()
ax.scatter(X[:,[1]],X[:,[2]],X[:,[3]],c=W.labels.astype(np.float))

ax.set_xlabel('Sepal width')
ax.set_ylabel('Petal length')
ax.set_zlabel('Petal width')

plt.show()
```



In [ ]: