In [43]:
```
#sklearn
#https://scikit-learn.org/stable/

from sklearn.datasets import load_iris
iris = load_iris()
```

In [44]:
```
# store the feature matrix (X): input,  and response vector (y): output (pre labele
X = iris.data
y = iris.target

feature_names = iris.feature_names
target_names = iris.target_names

print("Feature names:", feature_names)
print("Target names:", target_names)
```

```
Feature names: ['sepal length (cm)', 'sepal width (cm)', 'petal length (cm)', 'petal
width (cm)']
Target names: ['setosa' 'versicolor' 'virginica']
```

In [45]:
```
#Split data into training and test sets
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2)

print(X_train.shape)
print(X_test.shape)
```

```
(120, 4)
(30, 4)
```

In [61]:
```
#KNN Classifier. Try changing the n_neighbors
from sklearn.neighbors import KNeighborsClassifier
knn = KNeighborsClassifier(n_neighbors=3)
knn.fit(X_train, y_train)

#Decision Tree
# from sklearn.tree import DecisionTreeClassifier
# knn = DecisionTreeClassifier()
# knn.fit(X_train, y_train)


#make prediction
y_pred = knn.predict(X_test)
keepdims:False
```

```
C:\Users\10339\anaconda3\lib\site-packages\sklearn\neighbors\_classification.py:228:
FutureWarning: Unlike other reduction functions (e.g. `skew`, `kurtosis`), the defaul
t behavior of `mode` typically preserves the axis it acts along. In SciPy 1.11.0, thi
s behavior will change: the default value of `keepdims` will become False, the `axis`
over which the statistic is taken will be eliminated, and the value None will no long
er be accepted. Set `keepdims` to True or False to avoid this warning.
  mode, _ = stats.mode(_y[neigh_ind, k], axis=1)
```

In [58]:
```
from sklearn import metrics
print(metrics.accuracy_score(y_test, y_pred))
```

```
0.9333333333333333
```

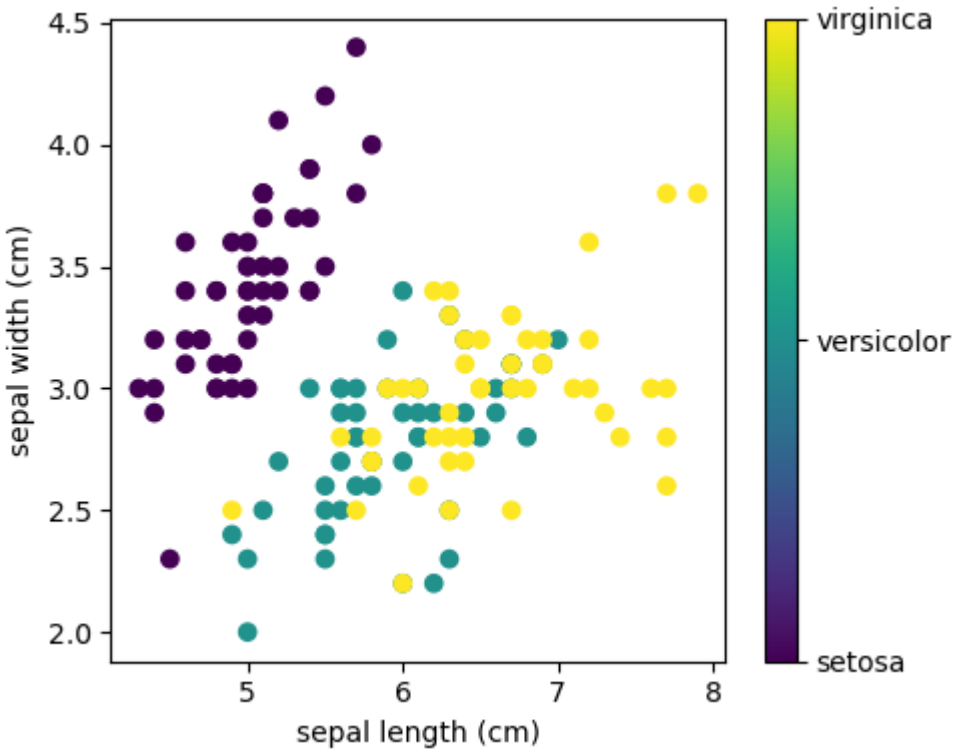
In [56]:
```
#Model persistance is important. Next time we want to make a prediction we save a mode
import joblib
from joblib import dump, load
joblib.dump(knn, 'mlbrain.joblib')
```

Out[56]:
```
['mlbrain.joblib']
```

In [52]:
```
#Load our model
model = joblib.load('mlbrain.joblib')


model.predict(X_test)
sample = [[3,5,4,2], [2,3,5,4]]
predictions = model.predict(sample)
pred_species = [iris.target_names[p] for p in predictions]
print("predictions: ", pred_species)
keepdims=False
```

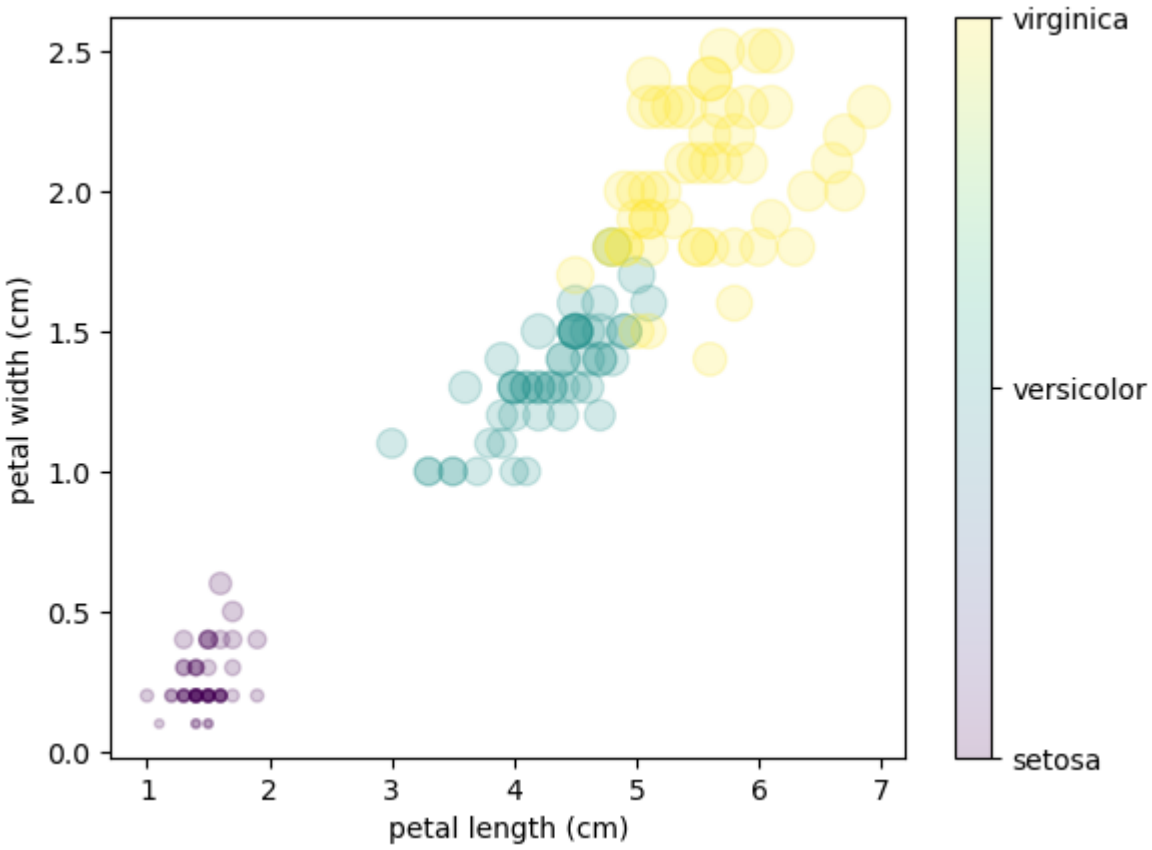
```
predictions:  ['versicolor', 'virginica']
C:\Users\10339\anaconda3\lib\site-packages\sklearn\neighbors\_classification.py:228:
FutureWarning: Unlike other reduction functions (e.g. `skew`, `kurtosis`), the defaul
t behavior of `mode` typically preserves the axis it acts along. In SciPy 1.11.0, thi
s behavior will change: the default value of `keepdims` will become False, the `axis`
over which the statistic is taken will be eliminated, and the value None will no long
er be accepted. Set `keepdims` to True or False to avoid this warning.
  mode, _ = stats.mode(_y[neigh_ind, k], axis=1)
C:\Users\10339\anaconda3\lib\site-packages\sklearn\neighbors\_classification.py:228:
FutureWarning: Unlike other reduction functions (e.g. `skew`, `kurtosis`), the defaul
t behavior of `mode` typically preserves the axis it acts along. In SciPy 1.11.0, thi
s behavior will change: the default value of `keepdims` will become False, the `axis`
over which the statistic is taken will be eliminated, and the value None will no long
er be accepted. Set `keepdims` to True or False to avoid this warning.
  mode, _ = stats.mode(_y[neigh_ind, k], axis=1)
```

In [28]:
```
from sklearn.datasets import load_iris
iris = load_iris()
import matplotlib.pyplot as plt

# The indices of the features that we are plotting
x_index = 0
y_index = 1

# colorbar with the Iris target names
formatter = plt.FuncFormatter(lambda i, *args: iris.target_names[int(i)])

#chart configurations
plt.figure(figsize=(5, 4))
plt.scatter(iris.data[:, x_index], iris.data[:, y_index], c=iris.target)
plt.colorbar(ticks=[0, 1, 2], format=formatter)
plt.xlabel(iris.feature_names[x_index])
plt.ylabel(iris.feature_names[y_index])

plt.tight_layout()
plt.show()
```



In [29]:
```
features = iris.data.T

plt.scatter(features[2], features[3], alpha=0.2,
            s=100*features[3], c=iris.target, cmap='viridis') #https://jakevdp.github.
plt.xlabel(iris.feature_names[2])
plt.ylabel(iris.feature_names[3]);
plt.colorbar(ticks=[0, 1, 2], format=formatter)
```

Out[29]:
```
<matplotlib.colorbar.Colorbar at 0x19229273f10>
```


petal width (cm)
petal length (cm)
0.0
0.5
1.0
1.5
2.0
2.5
1
2
3
4
5
6
7
virginica
versicolor
setosa

In [ ]: