Decision Tree

In [34]:

```
import pandas as pd
import matplotlib.pyplot as plt
from sklearn import datasets
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
from sklearn import tree
from sklearn.metrics import classification_report
from sklearn import preprocessing
```

In [35]:

```
#import some data to play with
Iris = pd.read_csv(r"C:\Users\HOME\Desktop\DSA\Lab11\iris.csv",index_col=0)
```

In [36]:

```
Iris.head()
```

Out[36]:

	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
1	5.1	3.5	1.4	0.2	setosa
2	4.9	3.0	1.4	0.2	setosa
3	4.7	3.2	1.3	0.2	setosa
4	4.6	3.1	1.5	0.2	setosa
5	5.0	3.6	1.4	0.2	setosa

In [37]:

```
Iris.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 150 entries, 1 to 150
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: 7.0+ KB
```

In [38]:

```
Iris.describe()
```

Out[38]:

	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width
count	150.000000	150.000000	150.000000	150.000000
mean	5.843333	3.057333	3.758000	1.199333
std	0.828066	0.435866	1.765298	0.762238
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 [39]:

```
#Complete Iris dataset
label_encoder = preprocessing.LabelEncoder()
Iris['Species']=label_encoder.fit_transform(Iris['Species'])
```

In [40]:

```
x = Iris.iloc[:,0:4]
y=Iris['Species']
```

In [41]:
```
Iris
```
Out[41]:

	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
1	5.1	3.5	1.4	0.2	0
2	4.9	3.0	1.4	0.2	0
3	4.7	3.2	1.3	0.2	0
4	4.6	3.1	1.5	0.2	0
5	5.0	3.6	1.4	0.2	0
...	...	...	...	...	...
146	6.7	3.0	5.2	2.3	2
147	6.3	2.5	5.0	1.9	2
148	6.5	3.0	5.2	2.0	2
149	6.2	3.4	5.4	2.3	2
150	5.9	3.0	5.1	1.8	2

150 rows × 5 columns

In [42]:
```
x
```
Out[42]:

	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width
1	5.1	3.5	1.4	0.2
2	4.9	3.0	1.4	0.2
3	4.7	3.2	1.3	0.2
4	4.6	3.1	1.5	0.2
5	5.0	3.6	1.4	0.2
...	...	...	...	...
146	6.7	3.0	5.2	2.3
147	6.3	2.5	5.0	1.9
148	6.5	3.0	5.2	2.0
149	6.2	3.4	5.4	2.3
150	5.9	3.0	5.1	1.8

150 rows × 4 columns

In [43]:
```
y
```
Out[43]:
```
1      0
2      0
3      0
4      0
5      0
      ..
146    2
147    2
148    2
149    2
150    2
Name: Species, Length: 150, dtype: int32
```

In [44]:
```
Iris['Species'].unique()
```
Out[44]:
```
array([0, 1, 2])
```

In [45]:
```
Iris.Species.value_counts()
```
Out[45]:
```
Species
0    50
1    50
2    50
Name: count, dtype: int64
```

In [46]:
```
colnames=list(Iris.columns)
```

In [47]:
```
colnames
```
Out[47]:
```
['Sepal.Length', 'Sepal.Width', 'Petal.Length', 'Petal.Width', 'Species']
```

In [48]:
```
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2,random_state=40)
```

Decision Tree Classifier

```
In [49]: model=DecisionTreeClassifier(criterion='entropy',max_depth=3)
         model.fit(x_train,y_train)
```

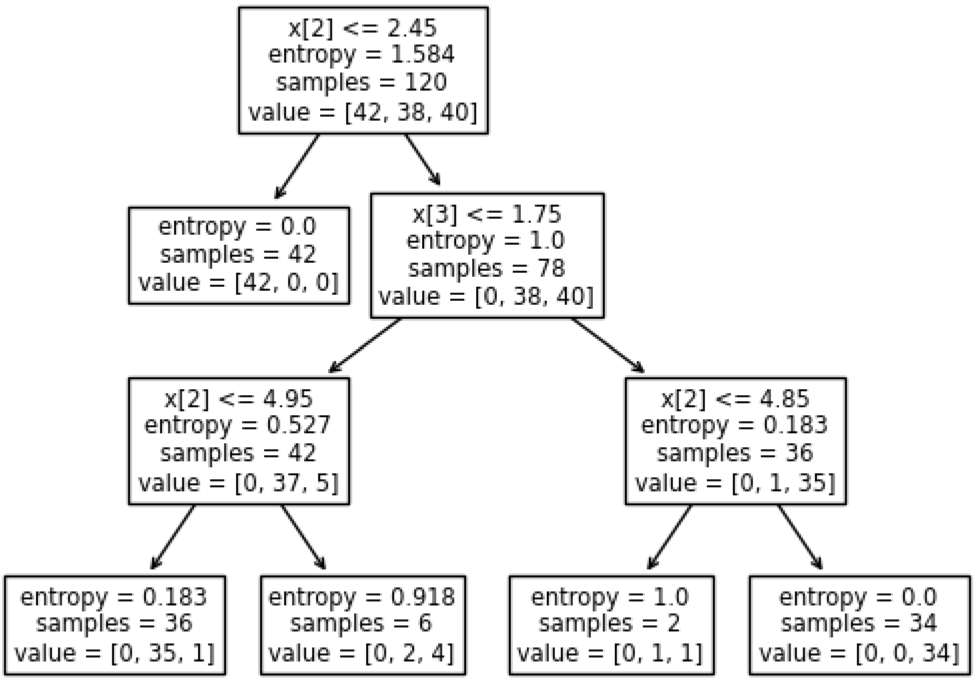
```
Out[49]: DecisionTreeClassifier(criterion='entropy', max_depth=3)
```

In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.
On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.

```
In [50]: #plot
```

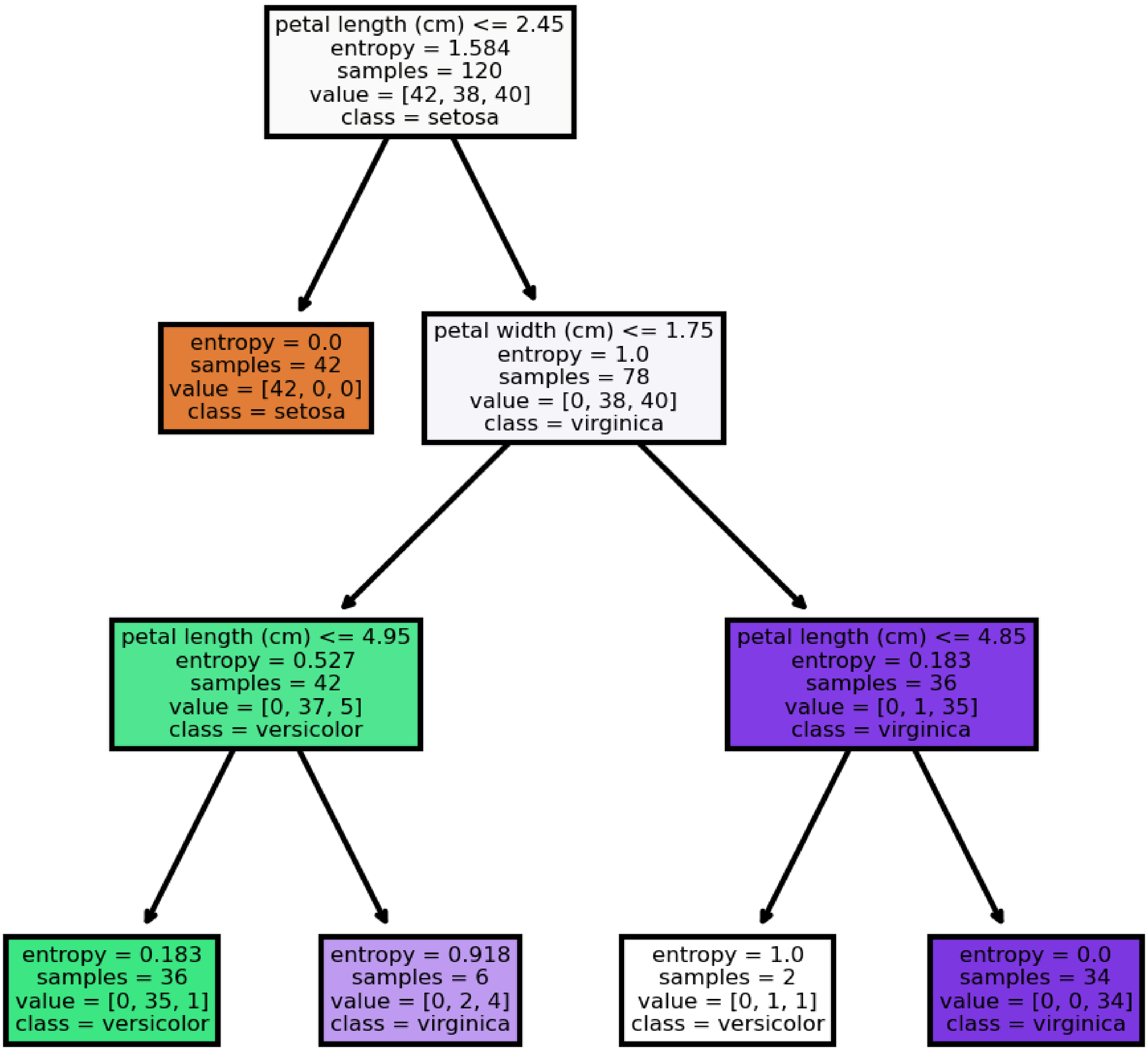
```
In [51]: tree.plot_tree(model)
```

```
Out[51]: [Text(0.375, 0.875, 'x[2] <= 2.45\nentropy = 1.584\nsamples = 120\nvalue = [42, 38, 40]'),
          Text(0.25, 0.625, 'entropy = 0.0\nsamples = 42\nvalue = [42, 0, 0]'),
          Text(0.5, 0.625, 'x[3] <= 1.75\nentropy = 1.0\nsamples = 78\nvalue = [0, 38, 40]'),
          Text(0.25, 0.375, 'x[2] <= 4.95\nentropy = 0.527\nsamples = 42\nvalue = [0, 37, 5]'),
          Text(0.125, 0.125, 'entropy = 0.183\nsamples = 36\nvalue = [0, 35, 1]'),
          Text(0.375, 0.125, 'entropy = 0.918\nsamples = 6\nvalue = [0, 2, 4]'),
          Text(0.75, 0.375, 'x[2] <= 4.85\nentropy = 0.183\nsamples = 36\nvalue = [0, 1, 35]'),
          Text(0.625, 0.125, 'entropy = 1.0\nsamples = 2\nvalue = [0, 1, 1]'),
          Text(0.875, 0.125, 'entropy = 0.0\nsamples = 34\nvalue = [0, 0, 34]')]
```



In [52]:

```
fn=['sepal length (cm)','sepal width (cm)','petal length (cm)','petal width (cm)']
cn=['setosa','versicolor','virginica']
fig,axes=plt.subplots(nrows=1,ncols=1,figsize=(4,4),dpi=300)
tree.plot_tree(model,
               feature_names=fn,
               class_names=cn,
               filled=True);
```



In [53]:

```
#predicting on test data
preds=model.predict(x_test)
pd.Series(preds).value_counts()
```

Out[53]:

```
1    13
2     9
0     8
Name: count, dtype: int64
```

In [54]:

```
preds
```

Out[54]:

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

In [55]:

```
pd.crosstab(y_test,preds)
```

Out[55]:

col_0	0	1	2
Species			
0	8	0	0
1	0	12	0
2	0	1	9

In [56]:

```
np.mean(preds==y_test)
```

Out[56]:

```
0.9666666666666667
```

Decision Tree CART using GINI Criteria

In [57]:

```
from sklearn.tree import DecisionTreeClassifier
model_gini = DecisionTreeClassifier(criterion='gini', max_depth=3)
```

In [58]:

```
model_gini.fit(x_train,y_train)
```

Out[58]:

```
DecisionTreeClassifier(max_depth=3)
```

In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.
On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.

In [59]:

```
#Prediction and Computing the accuracy
pred=model.predict(x_test)
np.mean(preds==y_test)
```

Out[59]:

```
0.9666666666666667
```

Decision Tree Regression

In [60]:

```
#Decision Tree Regression
from sklearn.tree import DecisionTreeRegressor
```

In [61]:

```
array = Iris.values
X = array[:,0:3]
y=array[:,3]
```

In [62]:

```
X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.33,random_state=1)
```

In [63]:

```
model=DecisionTreeRegressor()
model.fit(X_train,y_train)
```

Out[63]:

```
DecisionTreeRegressor()
```

In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.
On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.

In [64]:

```
model.score(X_test,y_test)
```

Out[64]:

```
0.8810631528394766
```