M5-L1 Problem 2

Now we will provide a 2D classification dataset and you will learn to use sklearn's decision tree classifier on the data.

First, run the following cell to load the data and import decision tree tools.

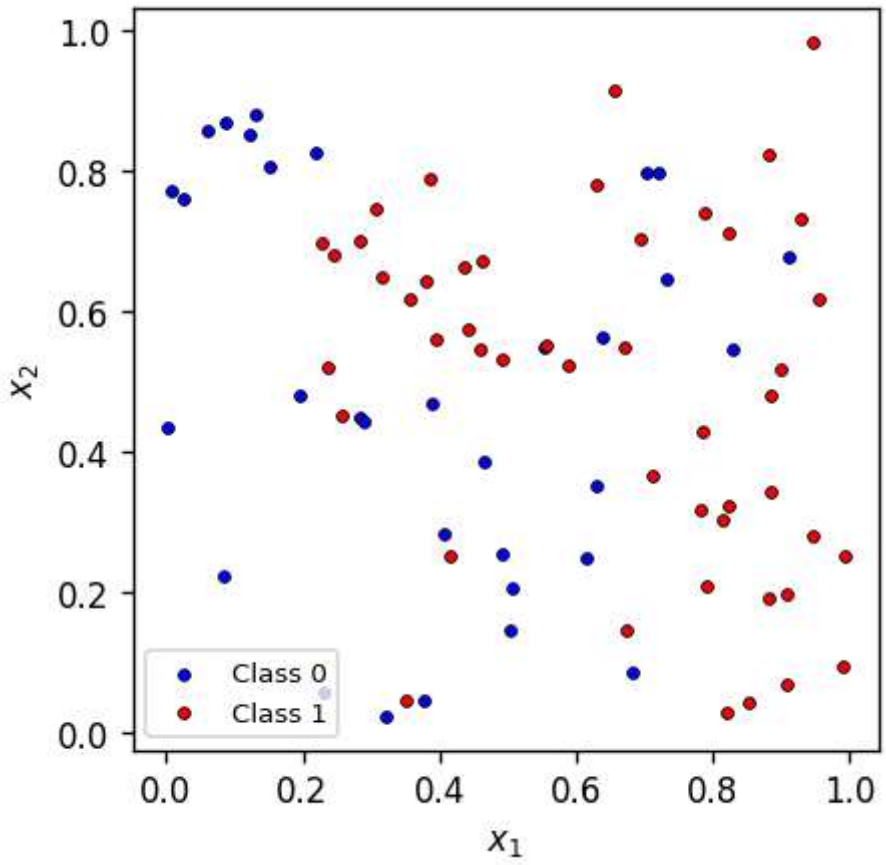
- Input: `X`, size 80×2
- Output: `y`, size 80

In [2]:
```
import numpy as np
import matplotlib.pyplot as plt
from sklearn.tree import DecisionTreeClassifier, plot_tree
from matplotlib.colors import ListedColormap

y = np.array([1, 1, 1, 0, 0, 1, 1, 0, 1, 1, 0, 1, 1, 1, 0, 1, 0, 0, 1, 1, 1, 1, 1,
x1 = np.array([6.73834679e-01, 3.57095269e-01, 4.42510505e-01, 8.48412660e-02, 2.17
x2 = np.array([0.14784469, 0.61647661, 0.57595235, 0.2232836 , 0.82559199, 0.545692
X = np.vstack([x1, x2]).T

def plot_data(X,y):
    colors=["blue","red"]
    for i in range(2):
        plt.scatter(X[y==i,0],X[y==i,1],s=12,c=colors[i],edgecolors="black",linewid
        plt.xlabel("$x_1$")
        plt.ylabel("$x_2$")
        plt.legend(loc="lower left",prop={'size':8})

plt.figure(figsize=(4,4),dpi=120)
plot_data(X,y)
plt.show()
```



Create and fit a decision tree classifier

Create an instance of a `DecisionTreeClassifier()` with `max_depth` of 5. Fit this to the data `X`, `y`.

For more details, consult: https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html

In [3]:
```
# YOUR CODE GOES HERE
dtf = DecisionTreeClassifier(max_depth=5)

dtf.fit(X,y)
```

Out[3]:
```
DecisionTreeClassifier(max_depth=5)
```

Making new predictions using your model

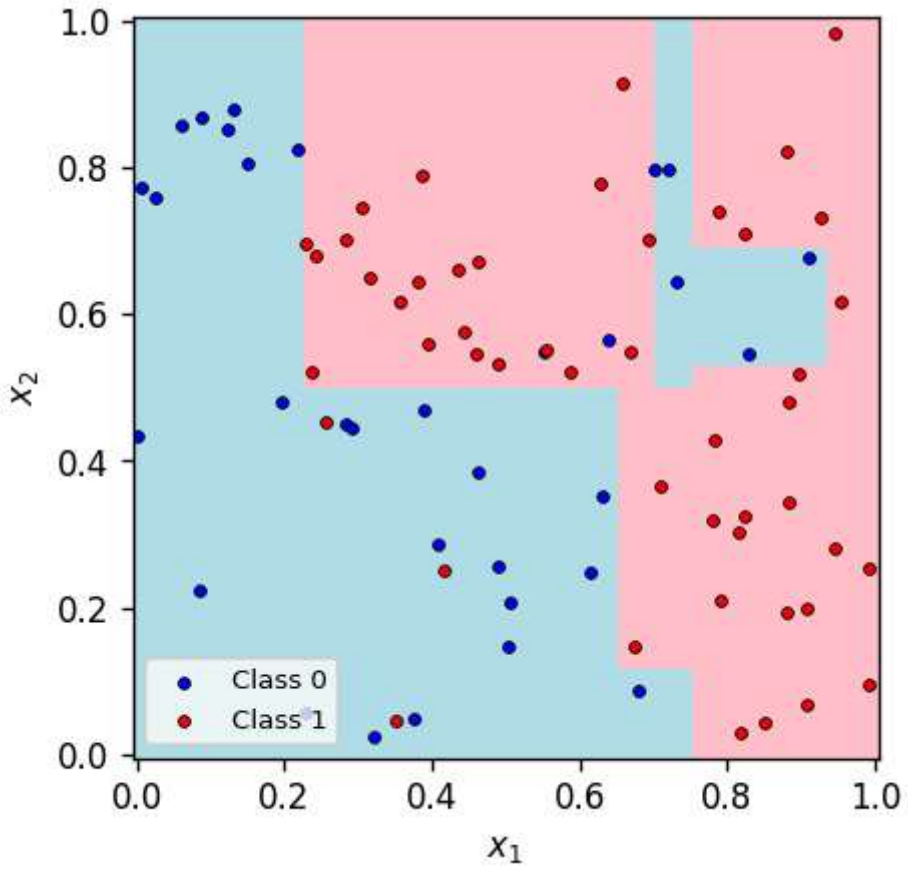
Now use the decision tree you trained to evaluate on the meshgrid of points `X_test` as indicated below. The code here will generate a plot showing the decision boundaries created by the model.

In [6]:
```
vals = np.linspace(0,1,100)
x1grid, x2grid = np.meshgrid(vals, vals)

X_test = np.vstack([x1grid.flatten(), x2grid.flatten()]).T


# YOUR CODE GOES HERE
# compute a prediction, `pred` for the input `X_test`
pred = dtf.predict(X_test)

plt.figure(figsize=(4,4),dpi=120)
bgcolors = ListedColormap(["powderblue","pink"])
plt.pcolormesh(x1grid, x2grid, pred.reshape(x1grid.shape), shading="nearest",cmap=b
plot_data(X,y)
plt.show()
```



Visualizing the decision tree

The `plot_tree()` function (https://scikit-learn.org/stable/modules/generated/sklearn.tree.plot_tree.html) can generate a simple visualization of your decision tree model. Try out this function below:

In [7]:
```
plt.figure(figsize=(4,4),dpi=250)

# YOUR CODE GOES HERE
```

```
plot_tree(dtf)
plt.show()
```