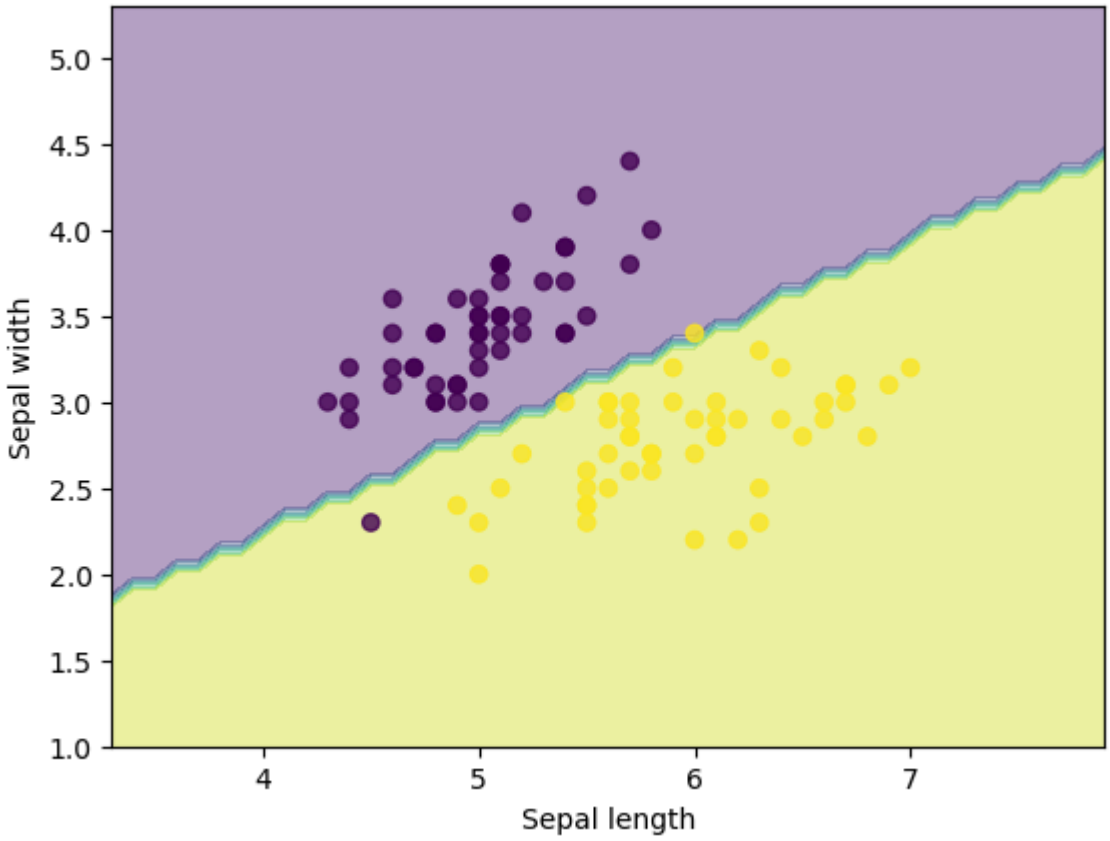
In [2]:
```
import numpy as np
from sklearn.datasets  import  load_iris

# Load the iris dataset
iris = load_iris()

X = iris.data[:100, :2]
y = iris.target[:100]

learning_rate = 0.1
num_iterations = 100

weights = np.random.rand(2)

def activation(x):
    return np.where(x >= 0, 1, 0)

for i in range(num_iterations):
    # Initialize the gradient and cost
    gradient = np.zeros(2)
    cost = 0

    # Loop over the training examples
    for j in range(len(X)):
        output = activation(np.dot(X[j], weights))
        error = y[j] - output

        # Update the gradient and cost
        gradient += error * X[j]
        cost += error ** 2

    weights += learning_rate * gradient

    print("Iteration:", i, "Cost:", cost)
```

```
Iteration: 0 Cost: 50
Iteration: 1 Cost: 50
Iteration: 2 Cost: 50
Iteration: 3 Cost: 50
Iteration: 4 Cost: 50
Iteration: 5 Cost: 50
Iteration: 6 Cost: 50
Iteration: 7 Cost: 50
Iteration: 8 Cost: 50
Iteration: 9 Cost: 50
Iteration: 10 Cost: 50
Iteration: 11 Cost: 50
Iteration: 12 Cost: 50
Iteration: 13 Cost: 50
Iteration: 14 Cost: 50
Iteration: 15 Cost: 50
Iteration: 16 Cost: 50
Iteration: 17 Cost: 50
Iteration: 18 Cost: 50
Iteration: 19 Cost: 47
Iteration: 20 Cost: 50
Iteration: 21 Cost: 45
Iteration: 22 Cost: 50
Iteration: 23 Cost: 46
Iteration: 24 Cost: 50
Iteration: 25 Cost: 45
Iteration: 26 Cost: 50
Iteration: 27 Cost: 45
Iteration: 28 Cost: 50
Iteration: 29 Cost: 45
Iteration: 30 Cost: 50
Iteration: 31 Cost: 45
Iteration: 32 Cost: 50
Iteration: 33 Cost: 45
Iteration: 34 Cost: 50
Iteration: 35 Cost: 45
Iteration: 36 Cost: 50
Iteration: 37 Cost: 46
Iteration: 38 Cost: 50
Iteration: 39 Cost: 45
Iteration: 40 Cost: 50
Iteration: 41 Cost: 45
Iteration: 42 Cost: 50
Iteration: 43 Cost: 45
Iteration: 44 Cost: 48
Iteration: 45 Cost: 44
Iteration: 46 Cost: 46
Iteration: 47 Cost: 42
Iteration: 48 Cost: 46
Iteration: 49 Cost: 42
Iteration: 50 Cost: 43
Iteration: 51 Cost: 31
Iteration: 52 Cost: 18
Iteration: 53 Cost: 3
Iteration: 54 Cost: 2
Iteration: 55 Cost: 2
Iteration: 56 Cost: 2
Iteration: 57 Cost: 1
Iteration: 58 Cost: 2
Iteration: 59 Cost: 2
Iteration: 60 Cost: 2
```

```
Iteration: 61 Cost: 1
Iteration: 62 Cost: 2
Iteration: 63 Cost: 2
Iteration: 64 Cost: 2
Iteration: 65 Cost: 1
Iteration: 66 Cost: 2
Iteration: 67 Cost: 2
Iteration: 68 Cost: 1
Iteration: 69 Cost: 2
Iteration: 70 Cost: 2
Iteration: 71 Cost: 2
Iteration: 72 Cost: 1
Iteration: 73 Cost: 2
Iteration: 74 Cost: 2
Iteration: 75 Cost: 2
Iteration: 76 Cost: 1
Iteration: 77 Cost: 2
Iteration: 78 Cost: 2
Iteration: 79 Cost: 2
Iteration: 80 Cost: 1
Iteration: 81 Cost: 2
Iteration: 82 Cost: 2
Iteration: 83 Cost: 1
Iteration: 84 Cost: 2
Iteration: 85 Cost: 2
Iteration: 86 Cost: 2
Iteration: 87 Cost: 1
Iteration: 88 Cost: 2
Iteration: 89 Cost: 2
Iteration: 90 Cost: 2
Iteration: 91 Cost: 1
Iteration: 92 Cost: 2
Iteration: 93 Cost: 2
Iteration: 94 Cost: 2
Iteration: 95 Cost: 1
Iteration: 96 Cost: 2
Iteration: 97 Cost: 2
Iteration: 98 Cost: 1
Iteration: 99 Cost: 2
```

In [3]:
```
import matplotlib.pyplot as plt
x_min, x_max = X[:, 0].min() - 1, X[:, 0].max() + 1
y_min, y_max = X[:, 1].min() - 1, X[:, 1].max() + 1
xx, yy = np.meshgrid(np.arange(x_min, x_max, 0.1),np.arange(y_min, y_max, 0.
Z = activation(np.dot(np.c_[xx.ravel(), yy.ravel()], weights))
Z = Z.reshape(xx.shape)
plt.contourf(xx, yy, Z, alpha=0.4)
plt.scatter(X[:, 0], X[:, 1], c=y, alpha=0.8)
plt.xlabel("Sepal length")
plt.ylabel("Sepal width")
plt.show()
```




In [ ]:
```

```