Assignment No.4

Title: Demonstrate the Perceptron learning law with its decision regions.

Name: Tavhare Ruchita Sharad

Roll No.: 58

In [1]:
```
import numpy as np
import matplotlib.pyplot as plt
from sklearn.datasets import load_iris
```

In [3]:
```
# load iris dataset
iris = load_iris()
```

In [7]:
```
# extract sepal length and petal length features
X = iris.data[:, [0, 2]]
y = iris.target
```

In [9]:
```
# setosa is class 0, versicolor is class 1
y = np.where(y == 0, 0, 1)

# initialize weights and bias
w = np.zeros(2)
b = 0
```

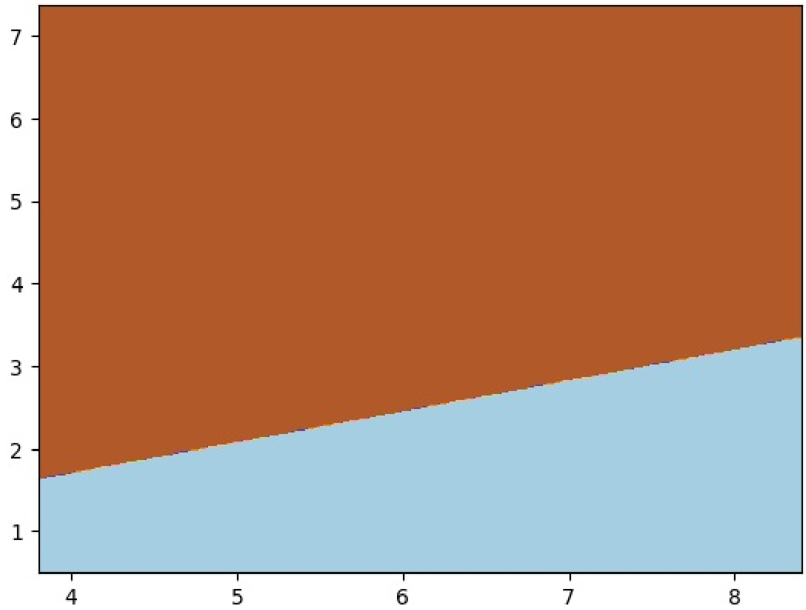
In [11]:
```
# set learning rate and number of epochs
lr = 0.1
epochs = 50

# define perceptron function
def perceptron(x, w, b):
    # calculate weighted sum of inputs
    z = np.dot(x, w) + b
    # apply step function
    return np.where(z >= 0, 1, 0)
```

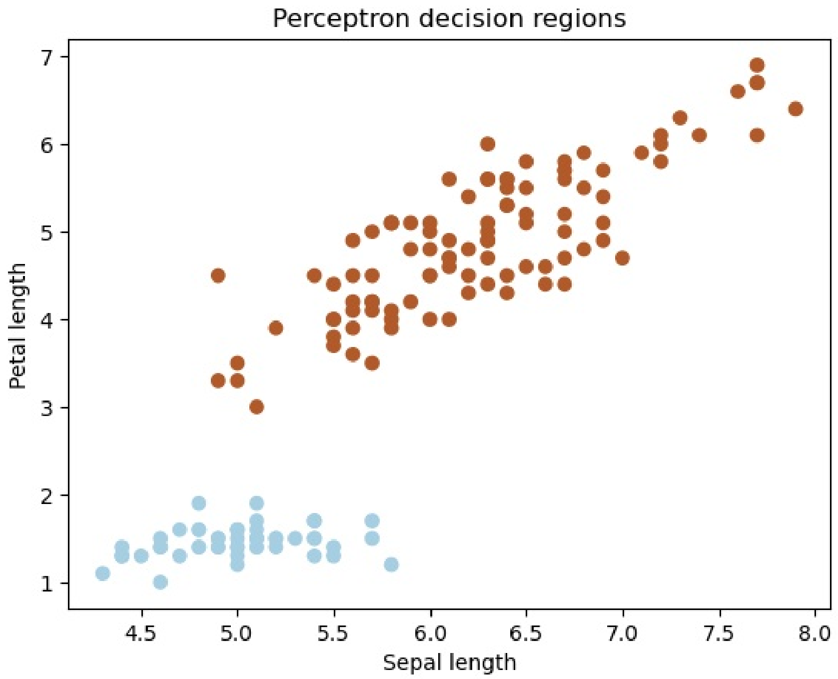
In [13]:
```
# train the perceptron
for epoch in range(epochs):
    for i in range(len(X)):
        x = X[i]
        target = y[i]
        output = perceptron(x, w, b)
        error = target - output
        w += lr * error * x
        b += lr * error
```

In [15]:
```
# plot decision boundary
x_min, x_max = X[:, 0].min() - 0.5, X[:, 0].max() + 0.5
y_min, y_max = X[:, 1].min() - 0.5, X[:, 1].max() + 0.5
xx, yy = np.meshgrid(np.arange(x_min, x_max, 0.02),np.arange(y_min, y_max, 0.02))
Z = perceptron(np.c_[xx.ravel(), yy.ravel()], w, b)
Z = Z.reshape(xx.shape)
plt.contourf(xx, yy, Z, cmap=plt.cm.Paired)
```

Out[15]:
```
<matplotlib.contour.QuadContourSet at 0x2b7e84169c0>
```



In [17]:
```
# plot data points
plt.scatter(X[:, 0], X[:, 1], c=y, cmap=plt.cm.Paired)
plt.xlabel('Sepal length')
plt.ylabel('Petal length')
plt.title('Perceptron decision regions')
plt.show()
```



In [ ]:

```

```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js