bl08r9srk

March 31, 2025

0.1 Practical 4

[3]:
```
import numpy as np
import matplotlib.pyplot as plt
from sklearn.linear_model import Perceptron
from matplotlib.colors import ListedColormap


X = np.array([[2, 3], [1, 1], [4, 3], [5, 5], [6, 6], [2, 5]])
y = np.array([0, 0, 1, 1, 1, 0])


perceptron = Perceptron(max_iter=1000, eta0=0.1, random_state=0)
perceptron.fit(X, y)


def plot_decision_regions(X, y, classifier, resolution=0.02):
    markers = ('s', 'o')
    colors = ('red', 'blue')
    cmap = ListedColormap(colors[:len(np.unique(y))])

    x1_min, x1_max = X[:, 0].min() - 1, X[:, 0].max() + 1
    x2_min, x2_max = X[:, 1].min() - 1, X[:, 1].max() + 1
    xx1, xx2 = np.meshgrid(np.arange(x1_min, x1_max, resolution),
                           np.arange(x2_min, x2_max, resolution))

    Z = classifier.predict(np.array([xx1.ravel(), xx2.ravel()]).T)
    Z = Z.reshape(xx1.shape)

    plt.contourf(xx1, xx2, Z, alpha=0.3, cmap=cmap)

    for idx, cl in enumerate(np.unique(y)):
        plt.scatter(x=X[y == cl, 0], y=X[y == cl, 1],
                    alpha=0.8, c=colors[idx], marker=markers[idx],
                    label=f'Class {cl}')


plot_decision_regions(X, y, perceptron)
```

```
plt.xlabel('Feature 1')
plt.ylabel('Feature 2')
plt.legend()
plt.title('Perceptron Decision Regions')
plt.show()
```