1.1 Closed-Form Solution

Given that the mean of the data is zero, i.e.,

$$\bar{\mathbf{x}} = \frac{1}{n}\sum_{i=1}^{n} \mathbf{x}_i = \mathbf{0},$$

we have:

$$\mathbf{1}^\top \mathbf{X} = n[\bar{x}_1, \bar{x}_2, \ldots, \bar{x}_m] = \mathbf{0}.$$

The objective function to minimize is:

$$J(\mathbf{w}, w_0) = (\mathbf{y}^\top - \mathbf{w}^\top \mathbf{X}^\top - w_0 \mathbf{1}^\top)(\mathbf{y} - \mathbf{X}\mathbf{w} - w_0\mathbf{1}) + \lambda \mathbf{w}^\top \mathbf{w}.$$

Taking the derivative of J with respect to w_0:

$$\frac{\partial J}{\partial w_0} = \frac{\partial}{\partial w_0}\left(-w_0\mathbf{y}^\top\mathbf{1} + w_0\mathbf{w}^\top\mathbf{X}^\top\mathbf{1} - w_0\mathbf{1}^\top\mathbf{y} + w_0\mathbf{1}^\top\mathbf{X}\mathbf{w} + w_0^2\mathbf{1}^\top\mathbf{1}\right).$$

Simplifying, we get:

$$\frac{\partial J}{\partial w_0} = \frac{\partial}{\partial w_0}\left(-2w_0 n\bar{\mathbf{y}} + nw_0^2\right) = -2n\bar{\mathbf{y}} + 2nw_0.$$

Setting the derivative to zero:

$$-2n\bar{\mathbf{y}} + 2nw_0 = 0 \implies \hat{w_0} = \bar{\mathbf{y}}.$$

Taking the derivative of J with respect to $\mathbf{w}$:

$$\frac{\partial J}{\partial \mathbf{w}} = \frac{\partial}{\partial \mathbf{w}}\left(-\mathbf{y}^\top\mathbf{X}\mathbf{w} - \mathbf{w}^\top\mathbf{X}^\top\mathbf{y} + \mathbf{w}^\top\mathbf{X}^\top\mathbf{X}\mathbf{w} + \lambda\mathbf{w}^\top\mathbf{w}\right).$$

Simplifying, we get:

$$\frac{\partial J}{\partial \mathbf{w}} = \frac{\partial}{\partial \mathbf{w}}\left(-2\mathbf{y}^\top\mathbf{X}\mathbf{w} + \mathbf{w}^\top(\mathbf{X}^\top\mathbf{X} + \lambda\mathbf{I})\mathbf{w}\right) = -2\mathbf{X}^\top\mathbf{y} + 2(\mathbf{X}^\top\mathbf{X} + \lambda\mathbf{I})\mathbf{w}.$$

Setting the derivative to zero:

$$-2\mathbf{X}^\top\mathbf{y} + 2(\mathbf{X}^\top\mathbf{X} + \lambda\mathbf{I})\mathbf{w} = 0 \implies \hat{\mathbf{w}} = (\mathbf{X}^\top\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^\top\mathbf{y}.$$

1.2 Support Vector Machine

(i) The decision boundary is given by:

$$f(x) = \mathbf{w}^\top \phi(x) + b = 0$$

The vector $\mathbf{w}$ is orthogonal to the decision boundary. Since there are only two points, the vector $\phi(x_2) - \phi(x_1)$ should also be orthogonal to the boundary. Therefore, a vector parallel to $\mathbf{w}$ is:

$$\phi(x_2) - \phi(x_1) = [0, \sqrt{2}, 1]^\top$$

(ii) The margin is given by:

$$\text{margin} = \frac{1}{2} \cdot \text{distance}(\phi(x_2), \phi(x_1)) = \frac{1}{2} \cdot \sqrt{(1-1)^2 + (0-\sqrt{2})^2 + (0-1)^2} = \frac{\sqrt{3}}{2}$$

(iii) Since w is parallel to $[0, \sqrt{2}, 1]^\top$, set $w = k[0, \sqrt{2}, 1]^\top$, where k is a constant. The margin is:

$$\frac{1}{\|w\|} = \frac{1}{k\sqrt{0+2+1}} = \frac{1}{k\sqrt{3}} = \frac{\sqrt{3}}{2}$$

Therefore $k = \frac{2}{3}$. The margin is:

$$w = k[0, \sqrt{2}, 1]^\top = \frac{2}{3}[0, \sqrt{2}, 1]^\top$$

(iv) Since we have:

$$y_1(\mathbf{w}^\top \phi(x_1) + w_0) = 1$$

Substituting the known values:

$$-1 \left(\frac{2}{3}[0, \sqrt{2}, 1] \cdot [1, 0, 0] + w_0 \right) = 1$$

We get $w_0 = -1$.

(v) The discriminant function is:

$$f(x) = w_0 + \mathbf{w}^\top \phi(x)$$

Substituting w and w_0:

$$f(x) = -1 + \frac{2}{3}[0, \sqrt{2}, 1] \cdot [1, \sqrt{2}x, x^2]$$

$$f(x) = -1 + \frac{2}{3}(\sqrt{2} \cdot \sqrt{2}x + 1 \cdot x^2)$$

$$f(x) = -1 + \frac{2}{3}(2x + x^2)$$

2.1 Logistic Regression

2.1.1 Load Data

The required output is shown below.

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
0	1	5.1	3.5	1.4	0.2	Iris-setosa
1	2	4.9	3.0	1.4	0.2	Iris-setosa
2	3	4.7	3.2	1.3	0.2	Iris-setosa
3	4	4.6	3.1	1.5	0.2	Iris-setosa
4	5	5.0	3.6	1.4	0.2	Iris-setosa
5	6	5.4	3.9	1.7	0.4	Iris-setosa
6	7	4.6	3.4	1.4	0.3	Iris-setosa
7	8	5.0	3.4	1.5	0.2	Iris-setosa
8	9	4.4	2.9	1.4	0.2	Iris-setosa
9	10	4.9	3.1	1.5	0.1	Iris-setosa

Figure 1: First 10 Lines

```
[[0]
 [0]
 [0]
 [0]
 [0]
 [1]
 [1]
 [1]
 [1]
 [1]
 [2]
 [2]
 [2]
 [2]
 [2]]
```

Figure 2: Numeric Columns in Array Form

```
[[5.1 3.5]
 [4.9 3. ]
 [4.7 3.2]
 [4.6 3.1]
 [5.  3.6]
 [5.4 3.9]
 [4.6 3.4]
 [5.  3.4]
 [4.4 2.9]
 [4.9 3.1]
 [5.4 3.7]
 [4.8 3.4]
 [4.8 3. ]
 [4.3 3. ]
 [5.8 4. ]
 [5.7 4.4]
 [5.4 3.9]
 [5.1 3.5]
 [5.7 3.8]
 [5.1 3.8]
 [5.4 3.4]
 [5.1 3.7]
 [4.6 3.6]
 [5.1 3.3]
 [4.8 3.4]
...
 [2]
 [2]
 [2]
 [2]]
```

Figure 3: Two ndarray

2.1.2 Softmax, Cost, and Derivative Functions

```
array([[0.33333333, 0.33333333, 0.33333333],
       [0.01587624, 0.11731043, 0.86681333],
       [0.09003057, 0.24472847, 0.66524096]])
```

Figure 4: Softmax Output

```
np.float64(0.8256461600462744)
```

Figure 5: Cost Function Output

The cross-entropy loss function for multi-class classification is:

$$J(\mathbf{W}) = -\frac{1}{m}\sum_{i=1}^{m}\sum_{j=1}^{C}\left[\mathbb{I}(y_i = j)\log \mathbf{f}_{\mathbf{w}_j}(\mathbf{x}_i)\right],$$

where m is the number of samples, C is the number of classes.

Using the chain rule, the gradient is:

$$\frac{\partial J(\mathbf{W})}{\partial \mathbf{w}_j} = -\frac{1}{m}\sum_{i=1}^{m}\left[\frac{\mathbb{I}(y_i = j)}{\mathbf{f}_{\mathbf{w}_j}(\mathbf{x}_i)} \cdot \frac{\partial \mathbf{f}_{\mathbf{w}_j}(\mathbf{x}_i)}{\partial \mathbf{w}_j} + \sum_{c\neq j}^{C}\frac{\mathbb{I}(y_i = c)}{\mathbf{f}_{\mathbf{w}_c}(\mathbf{x}_i)} \cdot \frac{\partial \mathbf{f}_{\mathbf{w}_c}(\mathbf{x}_i)}{\partial \mathbf{w}_j}\right]. \tag{1}$$

Since we have:

$$\frac{\partial \mathbf{f}_{\mathbf{w}_j}(\mathbf{x}_i)}{\partial \mathbf{w}_j} = \mathbf{f}_{\mathbf{w}_j}(\mathbf{x}_i)(1 - \mathbf{f}_{\mathbf{w}_j}(\mathbf{x}_i))\mathbf{x}_i,$$

For $c \neq j$, let:

$$\mathbf{f}_{\mathbf{w}_c}(\mathbf{x}_i) = \frac{N}{D},$$

where $N = e^{\mathbf{w}_c^\top \mathbf{x}_i}$, $D = \sum_{k=1}^{C} e^{\mathbf{w}_k^\top \mathbf{x}_i}$.

Using the quotient rule:

$$\frac{\partial \mathbf{f}_{\mathbf{w}_c}(\mathbf{x}_i)}{\partial \mathbf{w}_j} = \frac{\frac{\partial N}{\partial \mathbf{w}_j} \cdot D - N \cdot \frac{\partial D}{\partial \mathbf{w}_j}}{D^2}. \tag{2}$$

Since $N = e^{\mathbf{w}_c^\top \mathbf{x}_i}$ does not depend on $\mathbf{w}_j$:

$$\frac{\partial N}{\partial \mathbf{w}_j} = 0.$$

The derivative of the D with respect to $\mathbf{w}_j$ is:

$$\frac{\partial D}{\partial \mathbf{w}_j} = \frac{\partial}{\partial \mathbf{w}_j}\left(\sum_{k=1}^{C} e^{\mathbf{w}_k^\top \mathbf{x}_i}\right) = e^{\mathbf{w}_j^\top \mathbf{x}_i}\mathbf{x}_i.$$

Substituting these into the (2):

$$\frac{\partial \mathbf{f}_{\mathbf{w}_c}(\mathbf{x}_i)}{\partial \mathbf{w}_j} = \frac{0 \cdot D - N \cdot e^{\mathbf{w}_j^\top \mathbf{x}_i}\mathbf{x}_i}{D^2} = -\frac{N \cdot e^{\mathbf{w}_j^\top \mathbf{x}_i}\mathbf{x}_i}{D^2} = -\mathbf{f}_{\mathbf{w}_c}(\mathbf{x}_i)\mathbf{f}_{\mathbf{w}_j}(\mathbf{x}_i)\mathbf{x}_i.$$

Substituting into (1):

$$\frac{\partial J(\mathbf{W})}{\partial \mathbf{w}_j} = -\frac{1}{m}\sum_{i=1}^{m}\left[\frac{\mathbb{I}(y_i = j)}{\mathbf{f}_{\mathbf{w}_j}(\mathbf{x}_i)} \cdot \mathbf{f}_{\mathbf{w}_j}(\mathbf{x}_i)(1 - \mathbf{f}_{\mathbf{w}_j}(\mathbf{x}_i))\mathbf{x}_i + \sum_{c\neq j}^{C}\frac{\mathbb{I}(y_i = c)}{\mathbf{f}_{\mathbf{w}_c}(\mathbf{x}_i)} \cdot (-\mathbf{f}_{\mathbf{w}_c}(\mathbf{x}_i)\mathbf{f}_{\mathbf{w}_j}(\mathbf{x}_i)\mathbf{x}_i)\right].$$

Simplifying the expression:

$$\frac{\partial J(\mathbf{W})}{\partial \mathbf{w}_j} = -\frac{1}{m}\sum_{i=1}^{m}\left[\mathbb{I}(y_i = j)(1 - \mathbf{f}_{\mathbf{w}_j}(\mathbf{x}_i))\mathbf{x}_i - \sum_{c\neq j}^{C}\mathbb{I}(y_i = c)\mathbf{f}_{\mathbf{w}_j}(\mathbf{x}_i)\mathbf{x}_i\right].$$

Since $\sum_{c=1}^{C} \mathbb{I}(y_i = c) = 1$, we can get the final expression:

$$\frac{\partial J(\mathbf{W})}{\partial \mathbf{w}_j} = \frac{1}{m} \sum_{i=1}^{m} \left(\mathbf{f}_{\mathbf{w}_j}(\mathbf{x}_i) - \mathbb{I}(y_i = j) \right) \mathbf{x}_i.$$

```
array([[-0.05 , -0.175,  0.225],
       [ 0.325, -0.375,  0.05 ],
       [ 0.275, -0.55 ,  0.275]])
```

Figure 6: Gradient Output

2.1.3 Gradient Descent

The required output is shown below.

```
[Epoch 1], Cost function: 1.4265
[Epoch 2], Cost function: 1.3192
[Epoch 3], Cost function: 1.2950
[Epoch 4], Cost function: 1.2800
[Epoch 5], Cost function: 1.2671
[Epoch 6], Cost function: 1.2550
[Epoch 7], Cost function: 1.2433
[Epoch 8], Cost function: 1.2318
[Epoch 9], Cost function: 1.2205
[Epoch 10], Cost function: 1.2095
[Epoch 11], Cost function: 1.1987
[Epoch 12], Cost function: 1.1881
[Epoch 13], Cost function: 1.1777
[Epoch 14], Cost function: 1.1675
[Epoch 15], Cost function: 1.1575
[Epoch 16], Cost function: 1.1476
[Epoch 17], Cost function: 1.1380
[Epoch 18], Cost function: 1.1286
[Epoch 19], Cost function: 1.1194
[Epoch 20], Cost function: 1.1103
[Epoch 21], Cost function: 1.1015
[Epoch 22], Cost function: 1.0928
[Epoch 23], Cost function: 1.0843
[Epoch 24], Cost function: 1.0760
[Epoch 25], Cost function: 1.0678
...
[Epoch 9997], Cost function: 0.4085
[Epoch 9998], Cost function: 0.4085
[Epoch 9999], Cost function: 0.4085
[Epoch 10000], Cost function: 0.4085
```

Figure 7: Cost Function Record

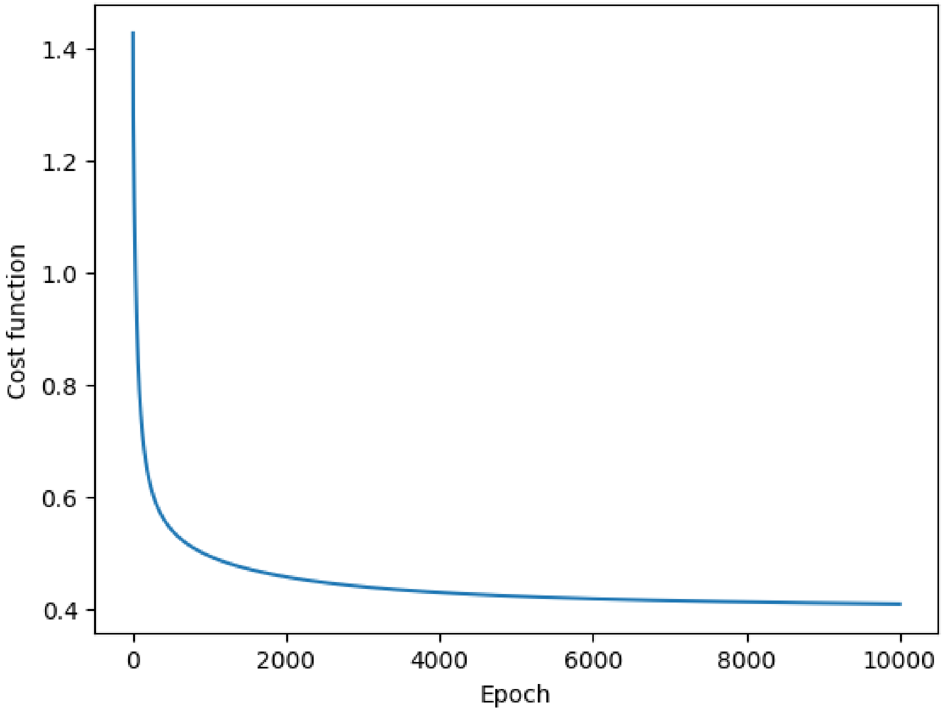

Figure 8: Cost Function v.s. Epoch

2.1.4 Plot Results

The required output is shown below.

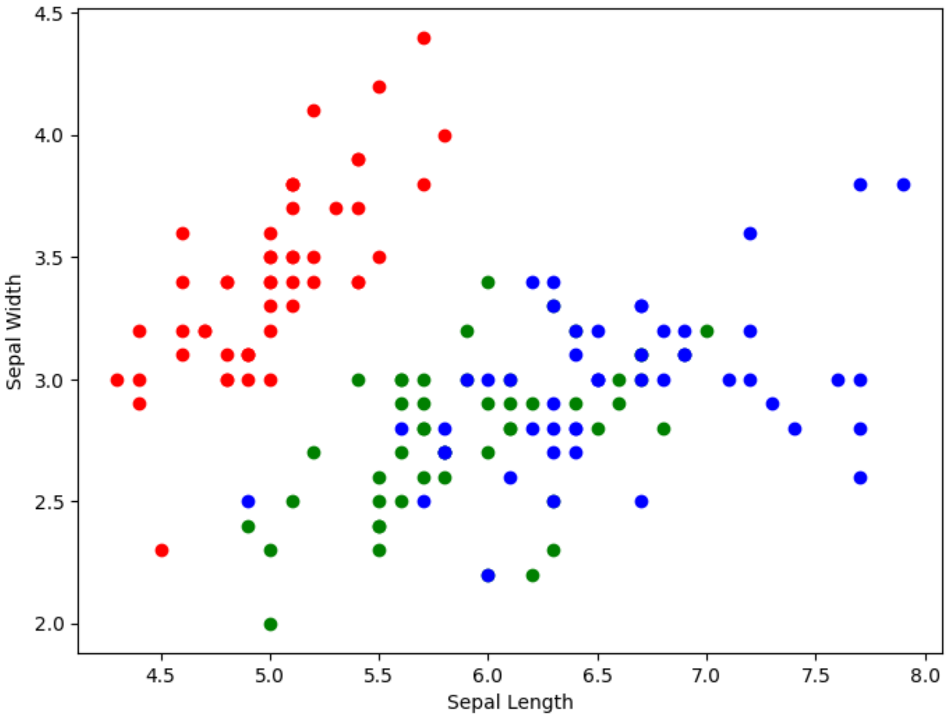

Figure 9: Ground Truth

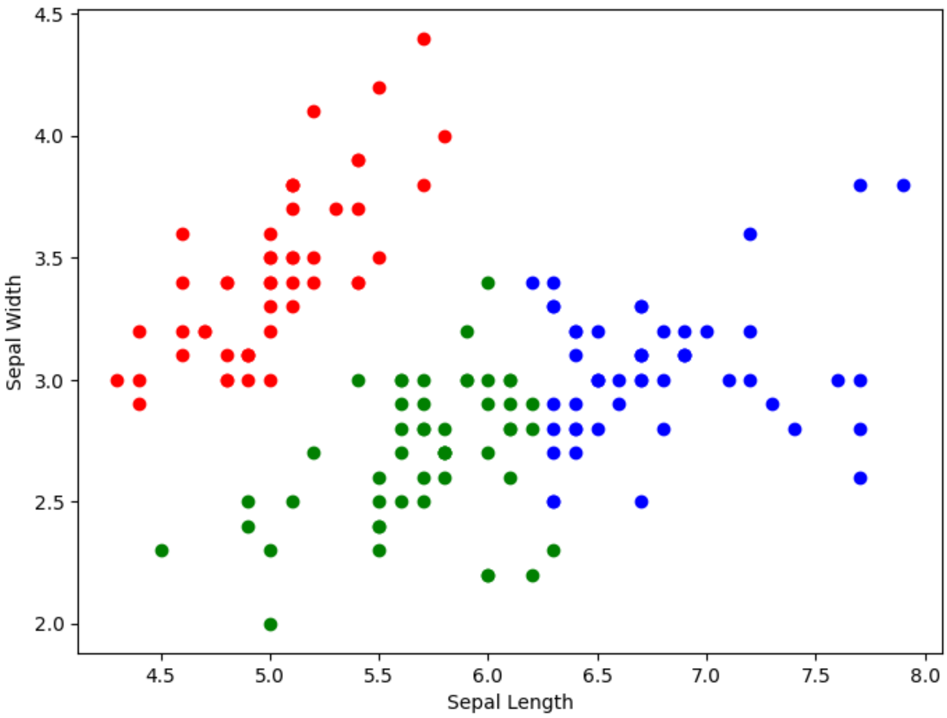

Figure 10: Predicted

2.2 SVM

The required output is shown below.

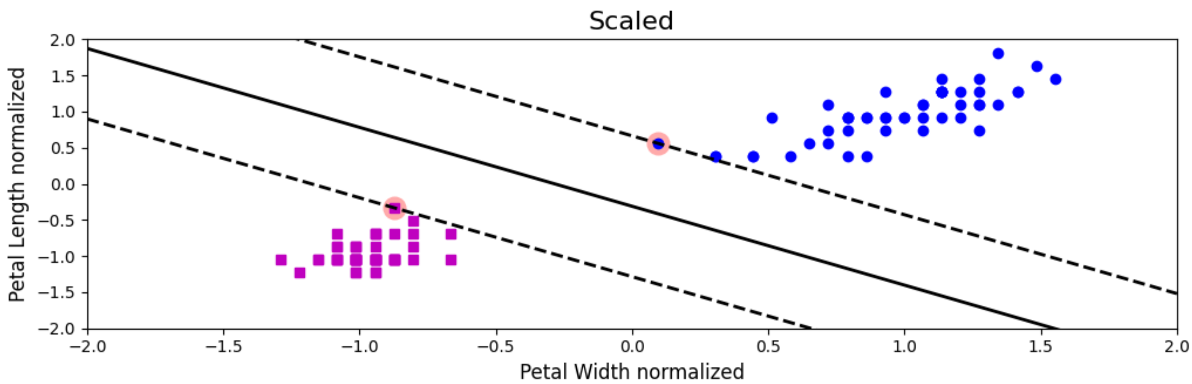

Figure 11: Linear SVM

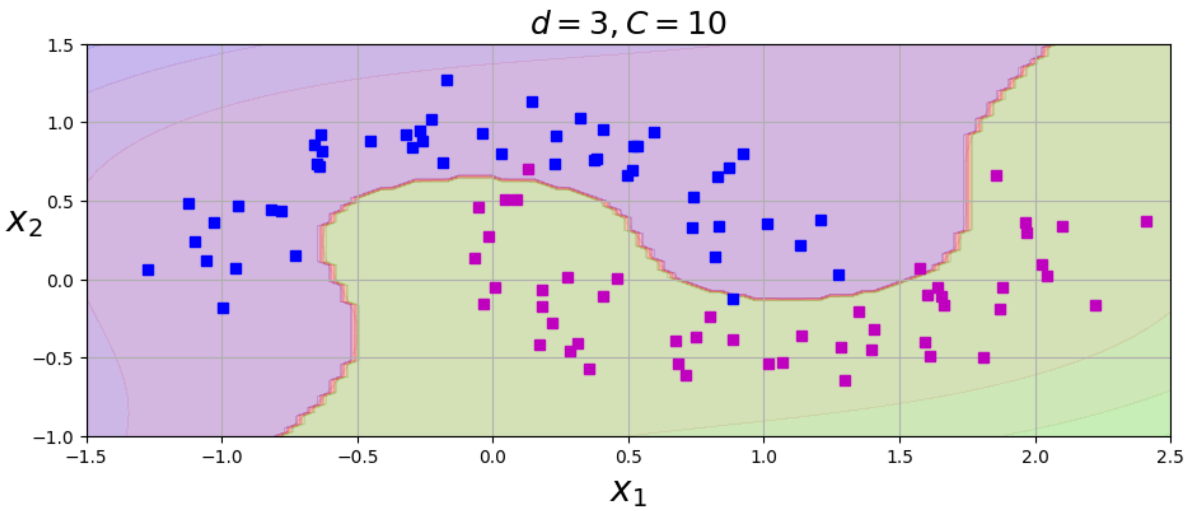

Figure 12: Non-Linear SVM