For the dimensionality reduction, we first consider the case of two classes, and then we can consider the generalization of the Linear discriminant for more than two classes.
Suppose we take the D-dimensional input vector X and project it down to one dimension using

$$y = W^T X$$

If we classify $y > -w_0$ as C_1, and otherwise as class C_2, then we obtain the standard linear classifier. We can, by adjusting the components of the weight vector W, select a projection that maximizes the class separation.
Consider that, there are N_1 examples of class C_1 and N_2 examples of class C_2, so the mean vectors of the two classes are given by

$$\mathbf{m_1} = \frac{\mathbf{1}}{\mathbf{N_1}} \sum_{\mathbf{n} \in C_1} \mathbf{X_n}, \qquad \mathbf{m_1} = \frac{\mathbf{1}}{\mathbf{N_2}} \sum_{\mathbf{n} \in C_2} \mathbf{X_n}$$

One way to measure the separation of classes is find the separation of the projected class means, when projected onto W. This implies we choose W so as to maximize

$$m_2 - m_1 = W^T(\mathbf{m_2} - \mathbf{m_1})$$

where,

$$m_k = W^T \mathbf{m_k}$$

is the mean of the projected data from class C_k. However, this expression can be made large by increasing the magnitude of W.
To solve this problem, we could constrain W to to have unit length, so that $\sum_i w_i^2 = 1$.
Using a Lagrange multiplier to perform the constrained maximization, we then find that $W \propto (\mathbf{m_2} - \mathbf{m_1})$.
The within-class variance of the transformed data from class C_k is therefore given by

$$s_k^2 = \sum_{n \in C_k} (y_n - m_k)^2$$

where $y_n = W^T X_n$. We can define the total within-class variance for the whole data set to be simply $s_1^2 + s_2^2$. The Fisher criterion is defined to be the ratio of the between-class variance to the within-class variance and is given by

$$J(W) = \frac{(m_1 - m_2)^2}{s_1^2 + s_2^2}.$$

Now, rewriting equation gives us an explicit dependence on w

$$J(W) = \frac{W^T S_B W}{W^T S_W W}.$$

where S_B is the *between-class* covariance matrix and is given by

$$S_B = (\mathbf{m_2} - \mathbf{m_1})(\mathbf{m_2} - \mathbf{m_1})^{\mathbf{T}}$$

and S_W is the total *within-class* covariance matrix, given by

$$S_W = \sum_{n \in C_1} (\mathbf{X_n} - \mathbf{m_1})(\mathbf{X_n} - \mathbf{m_1})^{\mathbf{T}} + \sum_{\mathbf{n} \in C_2} (\mathbf{X_n} - \mathbf{m_2})(\mathbf{X_n} - \mathbf{m_2})^{\mathbf{T}}$$

Differentiating J(W) with respect to w, we find that J(w) is maximized when

$$(W^T S_B W) S_W W = (W^T S_W W) S_B W$$

From the above equation, we see that $S_B W$ is always in the direction of $(\mathbf{m_2} - \mathbf{m_1})$. Furthermore, we do not care about the magnitude of w, only its direction, and so we can drop the scalar factors $(w^T S_B W)$ and $(W^T S_W W)$. Multiplying both sides of (4.29) by S_W^{-1} we then obtain

$$W \propto S_W^{-1}(\mathbf{m_2} - \mathbf{m_1})$$

The result is known as Fishers linear discriminant, although strictly it is not a discriminant but rather a specific choice of direction for projection of the data down to one dimension.

Now for K>2 classes, we can define covariance matrices in the projected **y**-space

$$s_B = \sum_{k=1}^{K} N_k (\mu_k - \mu)(\mu_k - \mu)^T$$

and

$$s_W = \sum_{k=1}^{K} \sum_{n \in C_k} (y_n - \mu_k)(y_n - \mu_k)^T$$

where

$$\mu_k = \frac{1}{N_k} \sum_{n \in C_k} y, \qquad \frac{1}{N} \sum_{k=1}^{K} N_k \mu_k$$

There are now many possible choices of criterion. One example is given by

$$J(W) = Tr\{s_W^{-1} s_B\}.$$

This criterion can then be rewritten as an explicit function of the projection matrix W in the form

$$J(W) = Tr\{(W S_W W^T)^{-1} (W S_B W^T)\}.$$

The weight values are determined by those eigenvectors of $S_W^{-1} S_B$ that corresponds to the largest eigenvalues.

We can employ the inner product matrix for calculation, this inner product matrix is also named Gram Matrix.

$$G_{N,N} = \begin{pmatrix} X_1^T X_1 & X_1^T X_2 & \cdots & X_1^T X_N \\ X_2^T X_1 & X_2^T X_2 & \cdots & X_2^T X_N \\ \vdots & \vdots & \ddots & \vdots \\ X_N^T X_1 & X_N^T X_2 & \cdots & X_N^T X_N \end{pmatrix}$$

To extend LDA to non-linear mappings, the data can be mapped to a new feature space via some function ϕ.

$$X_{i,j} \longmapsto \phi(X_{i,j})$$

Now the Gram Matrix becomes

$$G_{N,N} = \begin{pmatrix} \phi(X_1^T)\phi(X_1) & \phi(X_1^T)\phi(X_2) & \cdots & \phi(X_1^T)\phi(X_N) \\ \phi(X_2)^T\phi(X_1) & \phi(X_2^T)\phi(X_2) & \cdots & \phi(X_2^T)\phi(X_N) \\ \vdots & \vdots & \ddots & \vdots \\ \phi(X_N^T)\phi(X_1) & \phi(X_N^T)\phi(X_2) & \cdots & \phi(X_N^T)\phi(X_N) \end{pmatrix}$$

The Trick is not to explicitly use ϕ as it can be computationally expensive. We can make use of a kernel function which is dot product of ϕ vectors. The kernel function is defined as

$$K(X_{i,j}, X_{i,j}) = \phi(X_{i,j})^T \phi(X_{m,n})$$

Using the Kernel function and getting the matrix without explicitly mapping to the higher dimension is known as the **Kernel-LDA**.
Using the matrix, we can get the eigen vectors with only significant values. These eigen vector is $\phi(u)$. Now,

$$y = \phi(u)^T M^T \phi(X)$$

We need to get y, but $M^T\phi(X)$ is nothing but the kernel matrix columns and this eliminates the need of ϕ function explicitly.

$$M^T\phi(X) = \begin{bmatrix} K(X_{11}, X) \\ K(X_{12}, X) \\ \vdots \\ K(X_{NN}, X) \end{bmatrix} = Z$$

Now,

$$y = \phi(u)^T Z$$

The y can be calculated using the above result.

Kernel trick simplifies the task to be done as the need for searching for ϕ function is eliminated.