

1. Least squares:

The least squares method is based on the criterion of minimizing mean squared error. Let's consider the 2-class case:

Let C_0 and C_1 denote the two classes. Thus, if $y(i) = 0$ then $X_i \in C_0$ and if $y(i) = 1$ then $X_i \in C_1$

Let n_0 and n_1 denote the number of examples(features) of each class. ($n = n_0 + n_1$)

For any W , y are the one-dimensional data that we get after projection.

$$y_i = w^T x_i$$

We now determine the parameter matrix w , by minimizing a sum-of-squares error function:

$$E_D[w] = \frac{1}{2} T_R(xw - T)^T(xw - T)$$

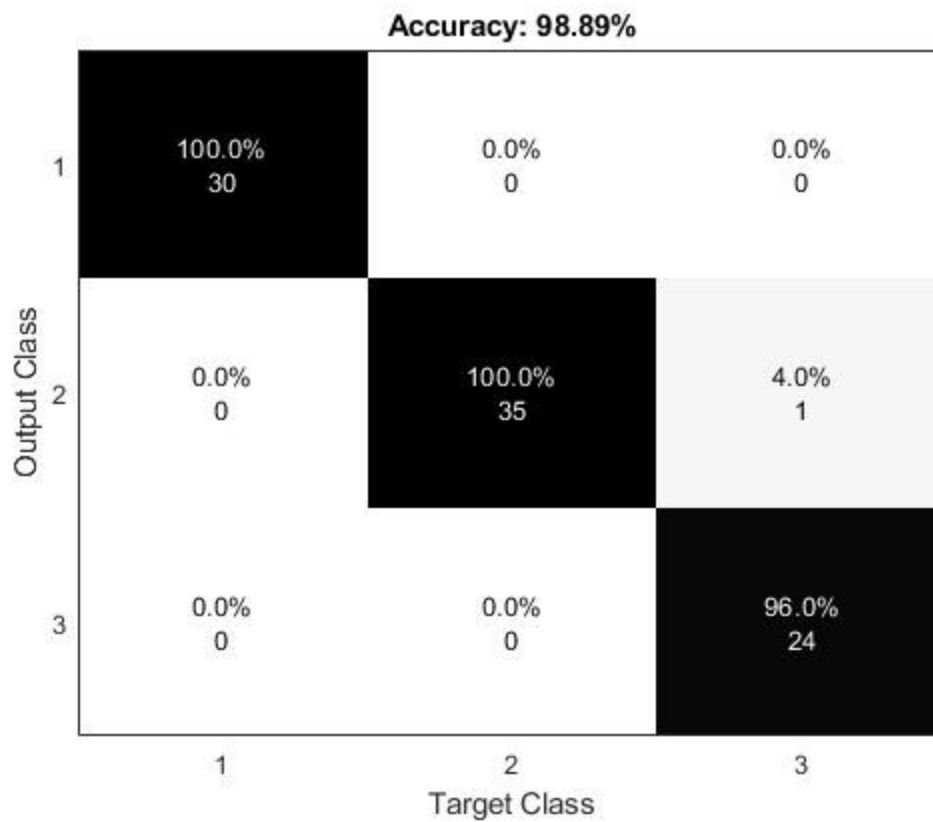
Setting the derivative with respect to w , to zero, and rearranging, we then obtain the solution for w , in the form:

$$w = (x^T x)^{-1} x^T T = x^\dagger T$$

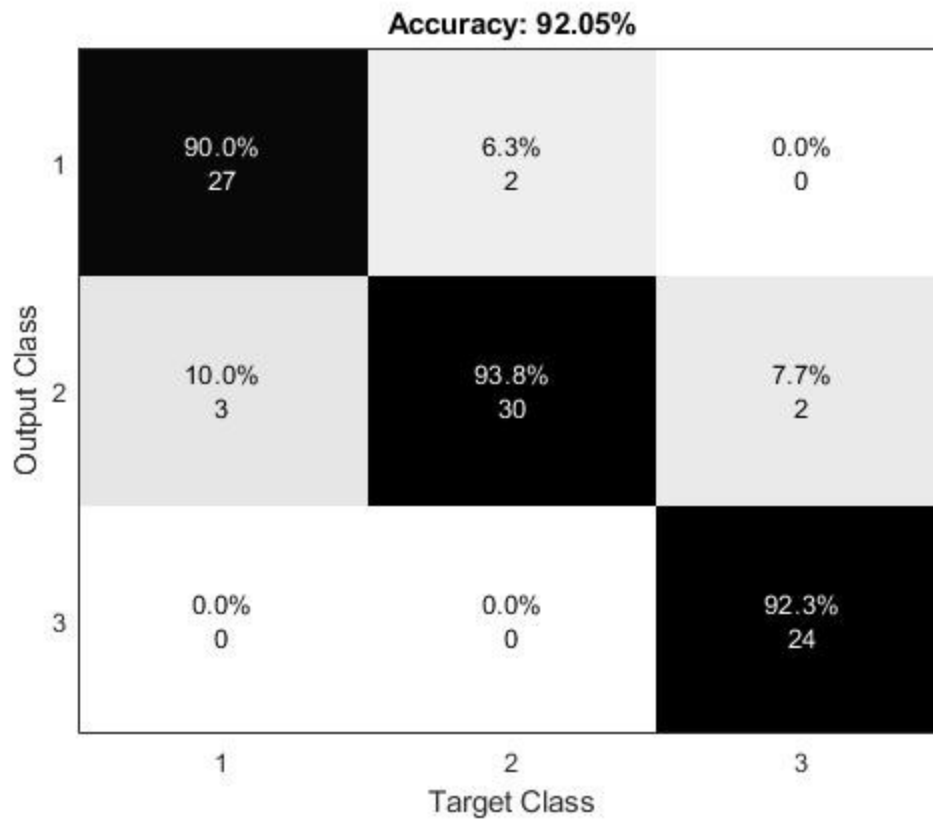
where x^\dagger is the pseudo-inverse of the matrix x . We then obtain the discriminant function in the form.

Following are the Confusion matrices for different datasets for both training and test sets:

a. Wine Data

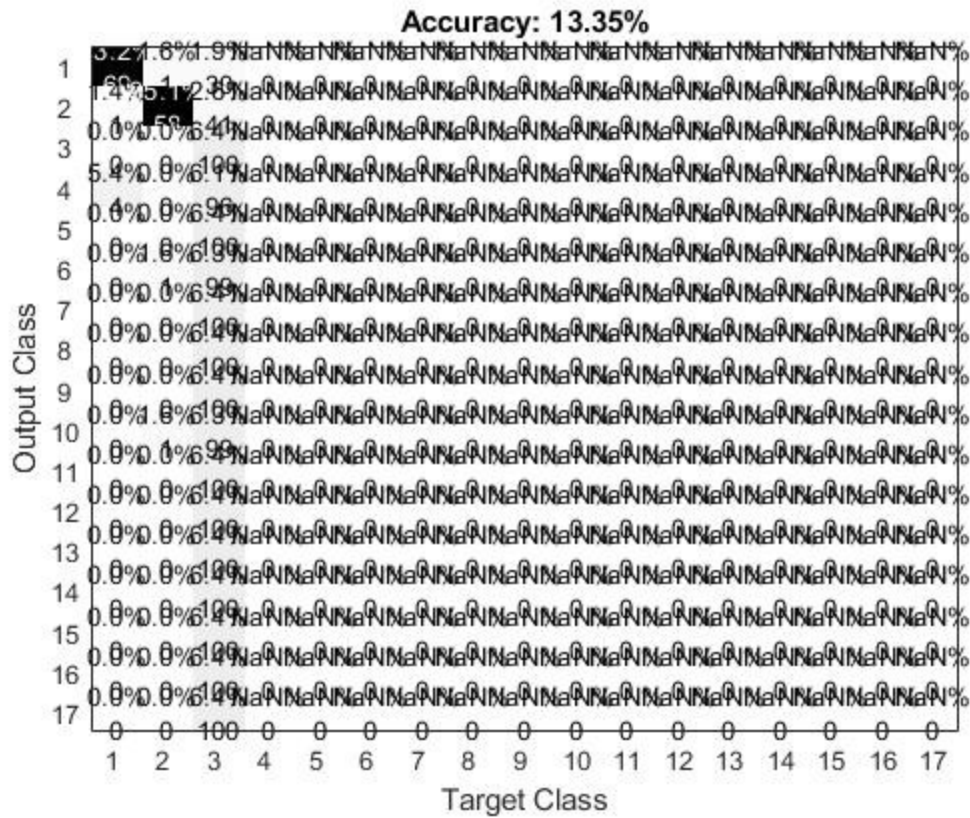


Training Data Set

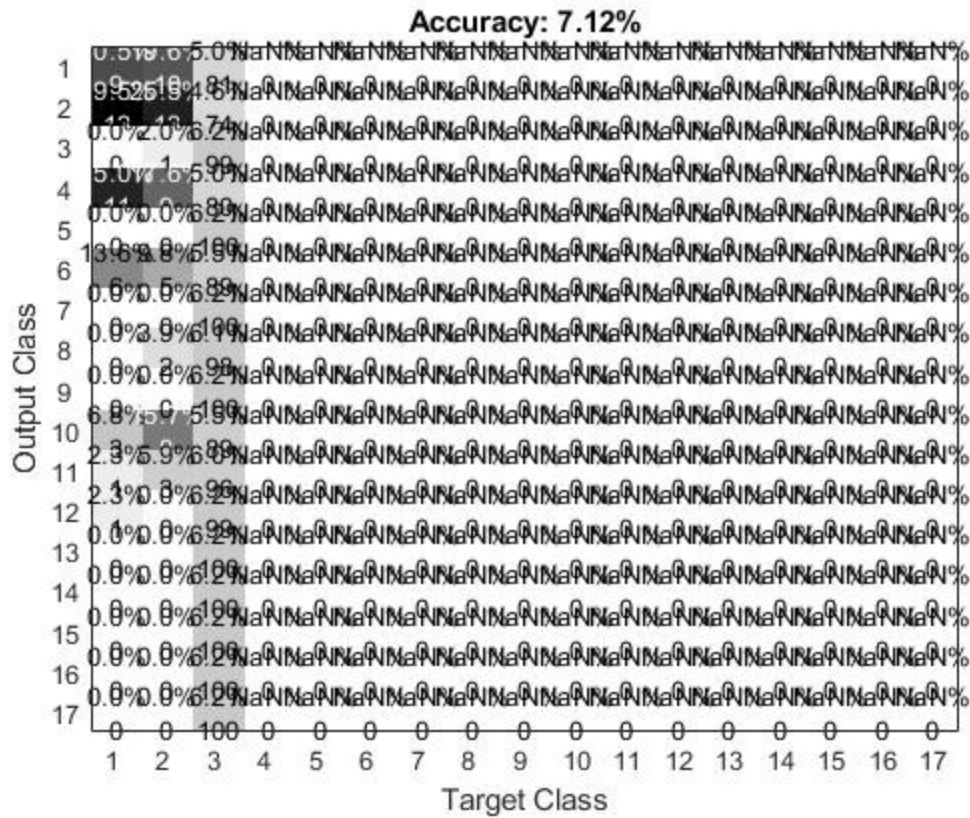


Test Data Set

b. Wallpaper Data



Training Data Set



Test Data Set

c. Taiji Data

Accuracy: 38.17%

1	99.9% 1139	5.5% 62	6.2% 566	NaN% 0	NaN% 0	NaN% 0	NaN% 0	NaN% 0
2	0.0% 0	94.5% 1066	0.0% 0	NaN% 0	NaN% 0	NaN% 0	NaN% 0	NaN% 0
3	0.0% 0	0.0% 0	23.4% 2132	NaN% 0	NaN% 0	NaN% 0	NaN% 0	NaN% 0
4	0.0% 0	0.0% 0	11.7% 1066	NaN% 0	NaN% 0	NaN% 0	NaN% 0	NaN% 0
5	0.0% 0	0.0% 0	11.7% 1066	NaN% 0	NaN% 0	NaN% 0	NaN% 0	NaN% 0
6	0.0% 0	0.0% 0	23.4% 2132	NaN% 0	NaN% 0	NaN% 0	NaN% 0	NaN% 0
7	0.0% 0	0.0% 0	11.7% 1066	NaN% 0	NaN% 0	NaN% 0	NaN% 0	NaN% 0
8	0.1% 1	0.0% 0	11.7% 1065	NaN% 0	NaN% 0	NaN% 0	NaN% 0	NaN% 0
	1	2	3	4	5	6	7	8

Target Class

Training Data Set

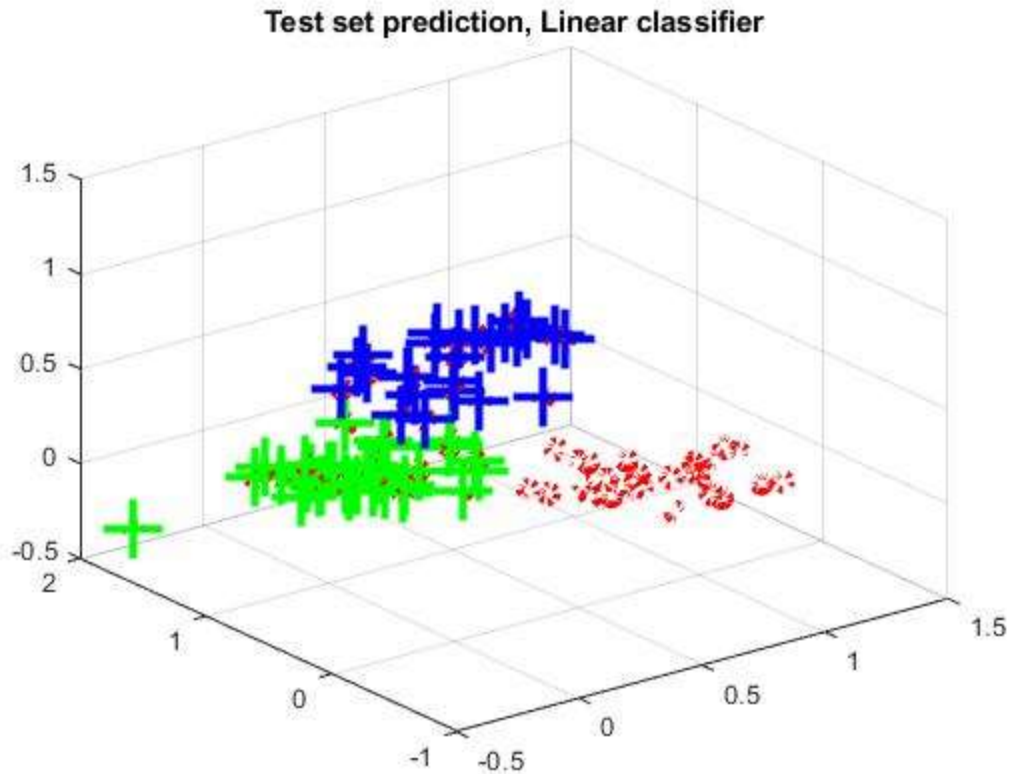
Accuracy: 34.30%

1	63.9% 239	7.3% 29	10.6% 335	NaN% 0	NaN% 0	NaN% 0	NaN% 0	NaN% 0
2	0.0% 0	92.7% 369	0.0% 0	NaN% 0	NaN% 0	NaN% 0	NaN% 0	NaN% 0
3	0.0% 0	0.0% 0	23.4% 738	NaN% 0	NaN% 0	NaN% 0	NaN% 0	NaN% 0
4	0.0% 0	0.0% 0	11.7% 369	NaN% 0	NaN% 0	NaN% 0	NaN% 0	NaN% 0
5	0.0% 0	0.0% 0	11.7% 369	NaN% 0	NaN% 0	NaN% 0	NaN% 0	NaN% 0
6	0.0% 0	0.0% 0	23.4% 738	NaN% 0	NaN% 0	NaN% 0	NaN% 0	NaN% 0
7	0.0% 0	0.0% 0	11.7% 369	NaN% 0	NaN% 0	NaN% 0	NaN% 0	NaN% 0
8	36.1% 135	0.0% 0	7.4% 234	NaN% 0	NaN% 0	NaN% 0	NaN% 0	NaN% 0
	1	2	3	4	5	6	7	8

Target Class

Test Data Set

Below is the 3-Dimensional figure for Wine testing:



2. Fisher LDA:

The least squares method is based on the criterion of minimizing mean squared error. Let's consider the 2-class case:

Let C_0 and C_1 denote the two classes. Thus, if $y(i) = 0$ then $X_i \in C_0$ and if $y(i) = 1$ then $X_i \in C_1$

Let n_0 and n_1 denote the number of examples(features) of each class. ($n = n_0 + n_1$)

For any W , z are the one-dimensional data that we get after projection.

$$z_i = w^T x_i$$

Let M_0 and M_1 be the means of data from the two classes:

$$M_0 = \frac{1}{n_0} \sum x_i$$

$$M_1 = \frac{1}{n_1} \sum x_i$$

we want a W that maximizes $[m_0 - m_1]^2$

However, we have to make this scale independent.

Also, the distance between means should be viewed relative to the variances.

Thus, we define:

$$s_0^2 = \sum_{x_i \in C_0} (w^T x_i - m_0)^2$$

$$s_1^2 = \sum_{x_i \in C_1} (w^T x_i - m_1)^2$$

These give us the variances (upto a factor) of the two classes in the projected data.

We want large separation between M_0 and M_1 relative to the variances.

Hence, we can take our objective to be to maximize:

$$J_w = \frac{(M_1 - M_0)^2}{s_0^2 + s_1^2}$$

$$(M_1 - M_0)^2 = [w^T M_1 - w^T M_0]^2$$

$$(M_1 - M_0)^2 = W^T (M_1 - M_0) (M_1 - M_0)^T W$$

$$= W^T S_B W$$

$$S_B = (M_1 - M_0) (M_1 - M_0)^T$$

S_B is a $d \times d$ matrix

We can similarly write s_0^2 and s_1^2 also as quadratic forms.

We know

$$s_0^2 = \sum_{x_i \in C_0} (w^T x_i - m_0)^2$$

$$s_1^2 = \sum_{x_i \in C_1} (w^T x_i - m_1)^2$$

$$s_0^2 + s_1^2 = W^T S_w W$$

S_w is also $d \times d$ matrix and is called within class scatter matrix

Thus,

$$J_w = \frac{W^T S_B W}{W^T S_w W}$$

We want to find a W that maximizes J_w :

J_w is not affected by scaling of W .

Maximizing ratio of quadratic forms is a standard optimization problem

Differentiating w.r.t. W and equating to zero, we get:

$$\frac{2S_B W}{W^T S_w W} - \frac{W^T S_B W}{W^T S_w W} * \frac{2 S_w W}{W^T S_w W} = 0$$

This Implies, $S_B W$ is in the same direction as $S_w W$

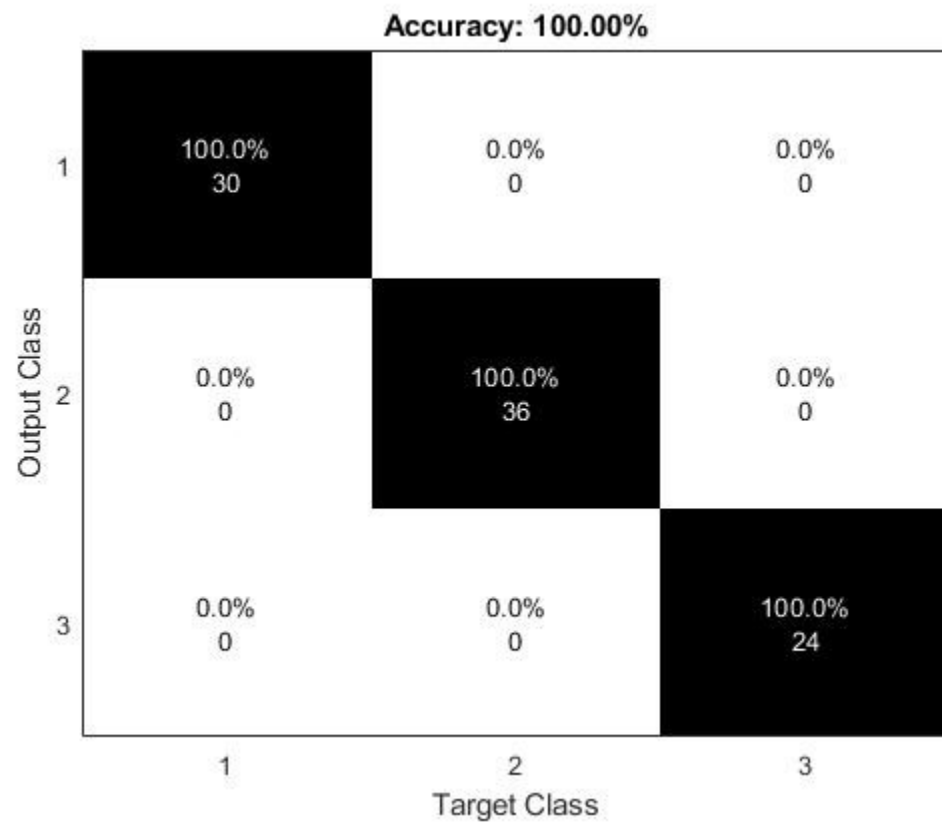
Thus, any maximizer of J_w has to satisfy $S_w W = \lambda S_b W$ for some constant λ

This is known as the generalized eigen value problem.

By solving the generalized eigen value problem we can find the best direction W .

Following are the Confusion matrices for different datasets for both training and test sets:

a) Wine dataset:

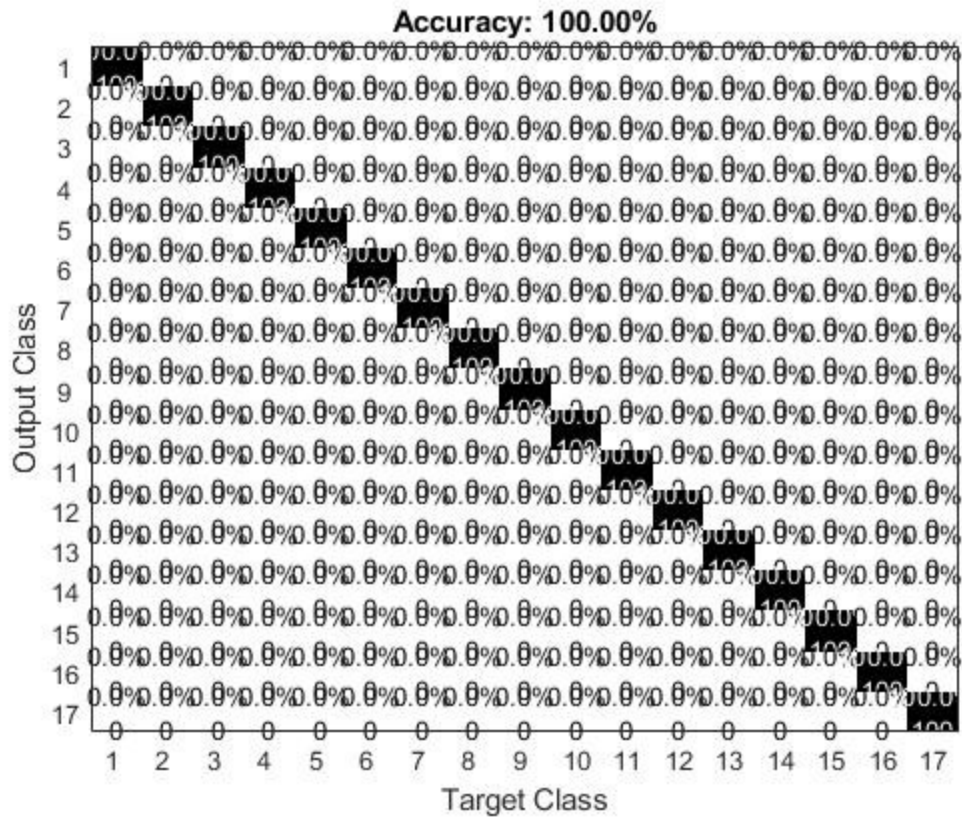


Training Data Set

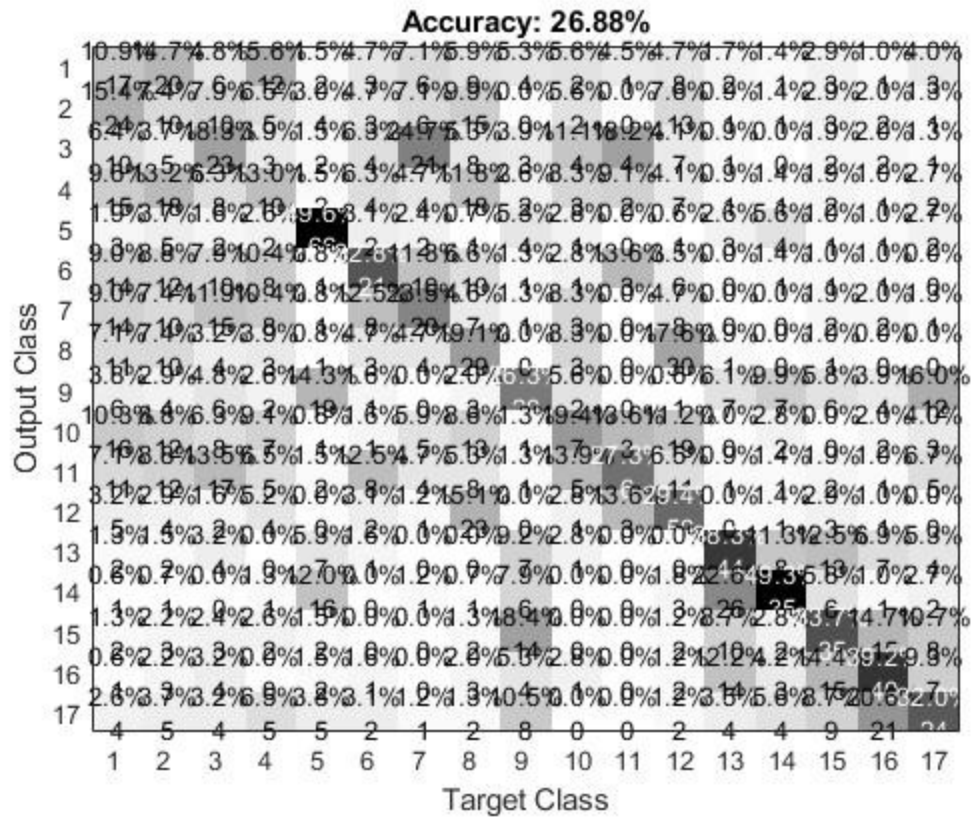


Test Data Set

b. Wallpaper Data Set:



Training Data Set



Test Data Set

c. **Taiji Data Set:**

Accuracy: 100.00%

1	100.0% 1767	0.0% 0	0.0% 0	0.0% 0	0.0% 0	0.0% 0	0.0% 0	0.0% 0
2	0.0% 0	100.0% 1066	0.0% 0	0.0% 0	0.0% 0	0.0% 0	0.0% 0	0.0% 0
3	0.0% 0	0.0% 0	100.0% 2132	0.0% 0	0.0% 0	0.0% 0	0.0% 0	0.0% 0
4	0.0% 0	0.0% 0	0.0% 0	100.0% 1066	0.0% 0	0.0% 0	0.0% 0	0.0% 0
5	0.0% 0	0.0% 0	0.0% 0	0.0% 0	100.0% 1066	0.0% 0	0.0% 0	0.0% 0
6	0.0% 0	0.0% 0	0.0% 0	0.0% 0	0.0% 0	100.0% 2132	0.0% 0	0.0% 0
7	0.0% 0	0.0% 0	0.0% 0	0.0% 0	0.0% 0	0.0% 0	100.0% 1066	0.0% 0
8	0.0% 0	0.0% 0	0.0% 0	0.0% 0	0.0% 0	0.0% 0	0.0% 0	100.0% 1066
	1	2	3	4	5	6	7	8

Target Class

Training Data Set

Accuracy: 48.90%

1	19.4% 318	24.7% 70	6.6% 47	12.5% 5	17.8% 62	4.2% 23	8.0% 16	38.0% 62
2	19.1% 313	17.3% 49	0.0% 0	0.0% 0	0.0% 0	0.0% 0	0.0% 0	4.3% 7
3	15.5% 253	0.0% 0	68.6% 485	0.0% 0	0.0% 0	0.0% 0	0.0% 0	0.0% 0
4	11.2% 184	43.5% 123	0.0% 0	52.5% 21	11.7% 41	0.0% 0	0.0% 0	0.0% 0
5	7.5% 123	0.0% 0	0.0% 0	0.0% 0	70.5% 246	0.0% 0	0.0% 0	0.0% 0
6	2.6% 42	0.0% 0	24.8% 175	0.0% 0	0.0% 0	95.8% 521	0.0% 0	0.0% 0
7	10.4% 170	0.0% 0	0.0% 0	35.0% 14	0.0% 0	0.0% 0	92.0% 185	0.0% 0
8	14.3% 234	14.5% 41	0.0% 0	0.0% 0	0.0% 0	0.0% 0	0.0% 0	57.7% 94
	1	2	3	4	5	6	7	8

Target Class

Test Data Set

3. Fisher LDA as special case of Linear Regression(for 2 classes):

Let's say the targets for class C1 be N/N_1 , where N_1 is the number of patterns in class C1, and N is the total number of patterns.

For class C_2 , we shall take the targets to be $-N/N_2$, where N_2 is the number of patterns in class C_2

The sum-of-squares error function can be written:

$$E = \frac{1}{2} \sum_{n=1}^N [w^T x_n + w_0 - t_n]^2$$

Setting the derivatives of E with respect to w_0 and \mathbf{w} to zero, we obtain respectively

$$\sum_1^N [w^T x_N + w_0 - t_n] x_N = 0$$

making use of our choice of target coding scheme for the t_n , we obtain an expression for the bias in the form.

$$\omega_0 = -w^T m$$

where we have used

$$\tilde{\Sigma} t_n = N_1 \frac{N}{N_1} - N_2 \frac{N}{N_2} = 0$$

and where \mathbf{m} is the mean of the total data set and is given by:

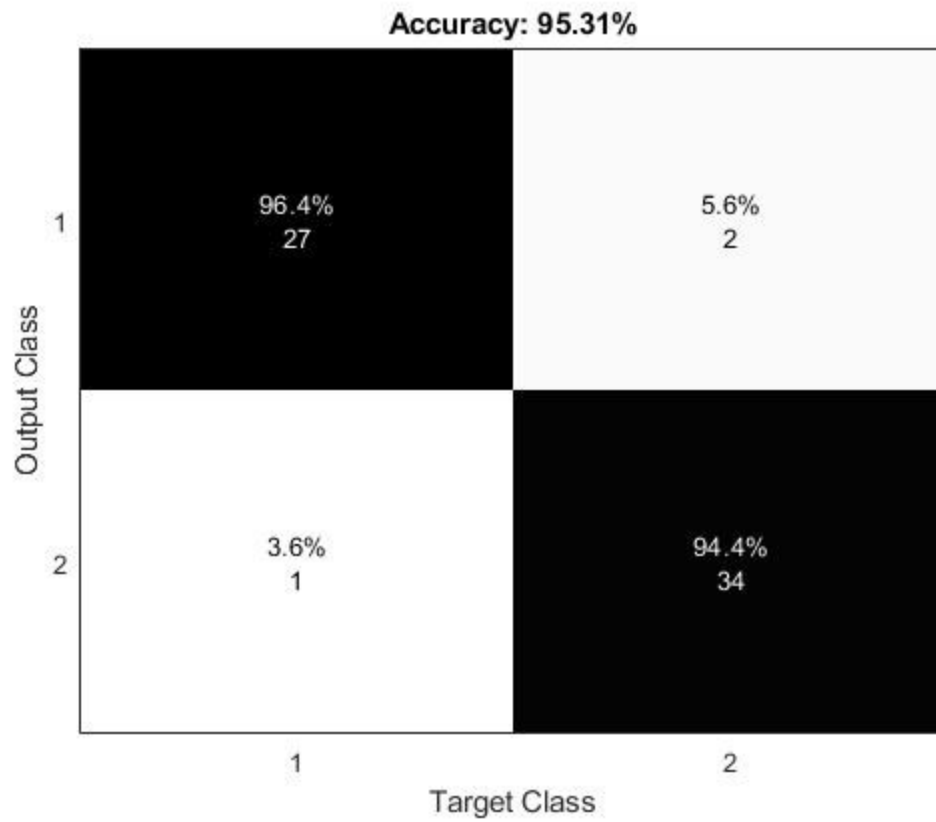
$$M = \frac{1}{N} (N_1 M_1 + N_2 M_2)$$

Thus we can write:

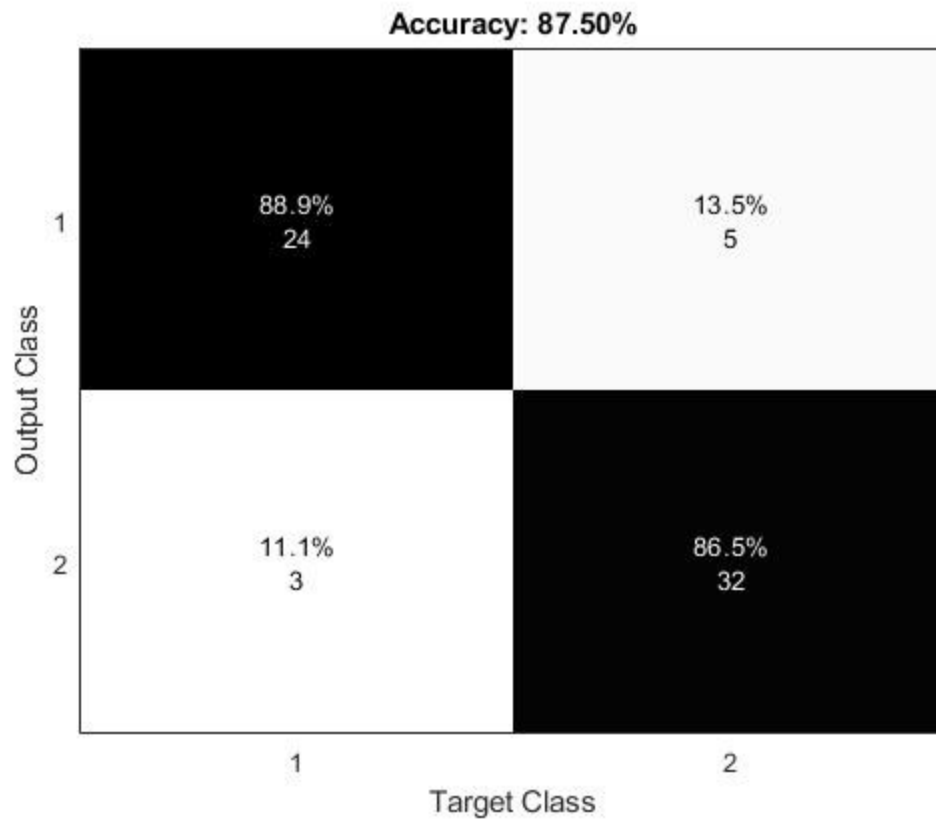
$$\left(S_{w + \frac{N_1 N_2}{N}} S_B \right) w = N(M_1 - M_2)$$

$$w \propto S_w^{-1} (M_2 - M_1)$$

Below are the confusion matrices for wine data after removing class 3:



Least Square Test Data Set



Fisher LDA Test Data Set