

# BITS F464 Machine Learning

## Fischer's Linear Discriminant

### 1. Introduction:

In this report, we try to analyse the performance (accuracy and fscore) of a Fischer's Linear Discriminant. The implementation of this model was done using Jupyter Notebook, NumPy, Pandas and Matplotlib.

### 2. Procedure:

1. The given dataset was initially split into two classes, positive and negative, according to the target variable.
2. The mean and covariance matrix for both classes were found and the overall covariance matrix  $S_W$  was obtained as the sum of covariance matrices of the positive and negative classes, i.e.  $S_W = S_1 + S_2$ .
3. The hyperplane onto which the points had to be collapsed was found using the equation  $W = S_W^{-1}(m_2 - m_1)$ , where  $m_2$  and  $m_1$  represented the means of the positive and negative classes.
4. Next, the points were reduced to 1D by taking a dot product of the direction ratios of the vectors (points in nD) and the direction ratios of the hyperplane ( $W$ ). The points after being reduced to 1D were plotted with positive points having red color and negative having blue color (as shown below).
5. The gaussian curves were plotted for both classes (with the reduced 1D feature as input) and the point of intersection was found by solving the resulting quadratic (say  $\alpha$ ).
6. Now, the point  $\alpha$  is the discriminating point in the reduced 1D space. The discriminating line passing through that point has been shown below. The differentiating hyperplane equation is found in the original feature space using  $W^T \alpha$ . A point was classified as negative if it was to the left of  $\alpha$  and positive if it was to the right of  $\alpha$ .
7. Finally, the confusion matrix for the model was generated and accuracy and fscore values were calculated. The model achieved 99.3% accuracy on dataset-1 and 100% accuracy on dataset-2 (results shown below).

### 3. Results:

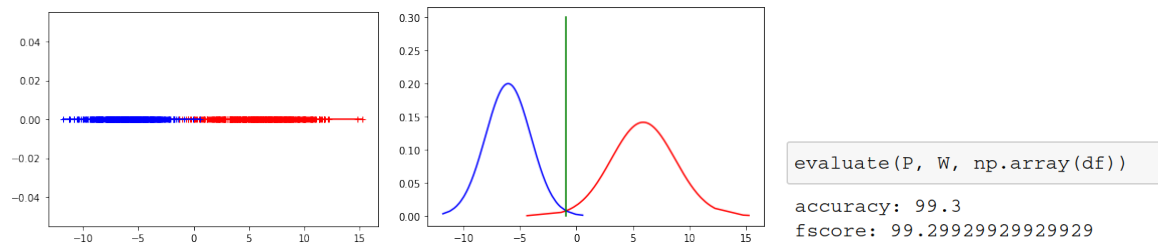


Figure 1: Dataset-I

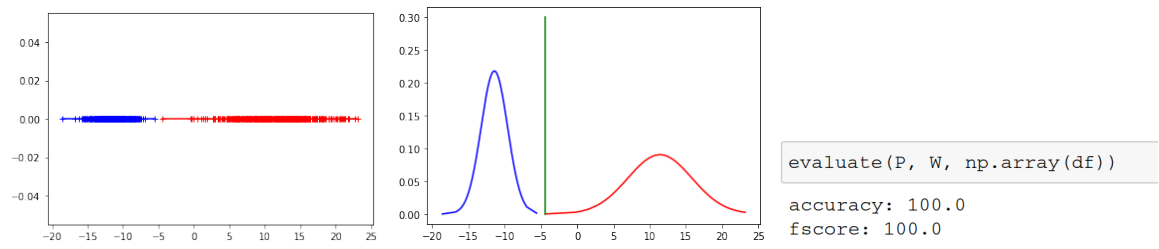


Figure 2: Dataset-II