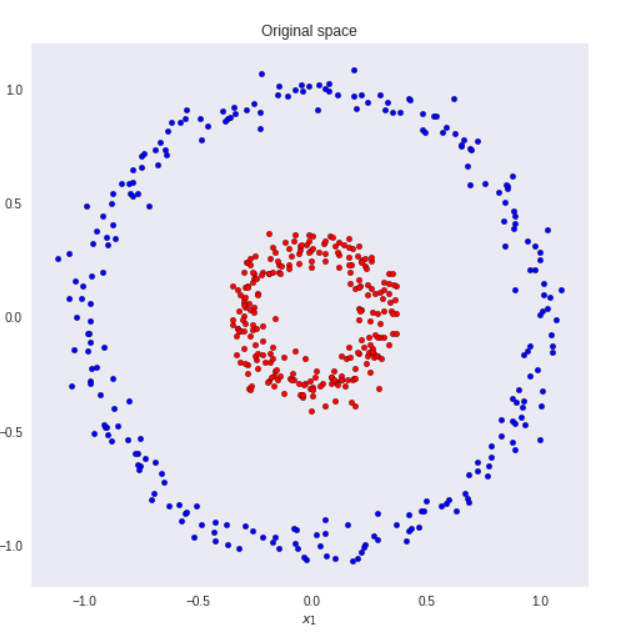
**Introduction**

To deal with classification problems with 2 or more classes, most Machine Learning (ML) algorithms work the same way.

Usually, they apply some kind of transformation to the input data with the effect of reducing the original input dimensions to a new (smaller) one. The goal is to project the data to a new space. Then, once projected, they try to classify the data points by finding a linear separation.

For problems with small input dimensions, the task is somewhat easier. Take the following dataset as an example.



## Fisher’s Linear Discriminant

**We can view linear classification models in terms of dimensionality reduction.**

To begin, consider the case of a two-class classification problem **(K=2)**. Blue and red points in R². In general, we can take any D-dimensional input vector and project it down to D’-dimensions. Here, **D** represents the original input dimensions while **D’** is the projected space

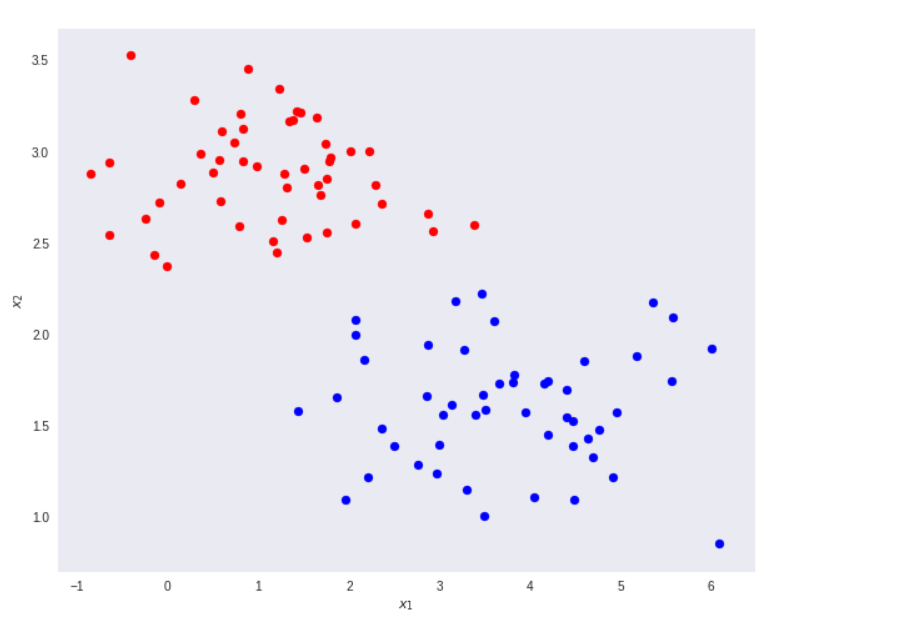
dimensions. Throughout this article, consider **D’** less than **D**.

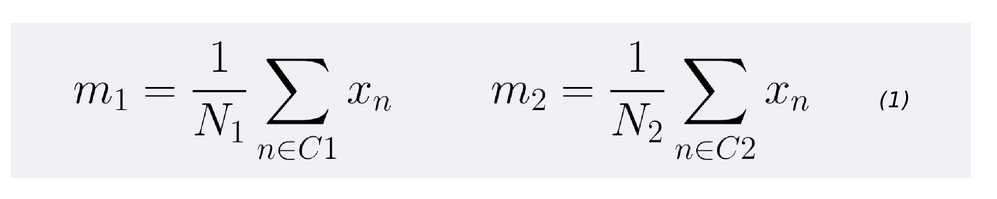
In the case of projecting to one dimension (the number line), i.e. **D’=1**, we can pick a threshold **t** to separate the classes in the new space. Given an input vector **x**:

* if the predicted value y >= t then, **x** belongs to class C1 (class 1) - where .
* otherwise, it is classified as C2 (class 2).

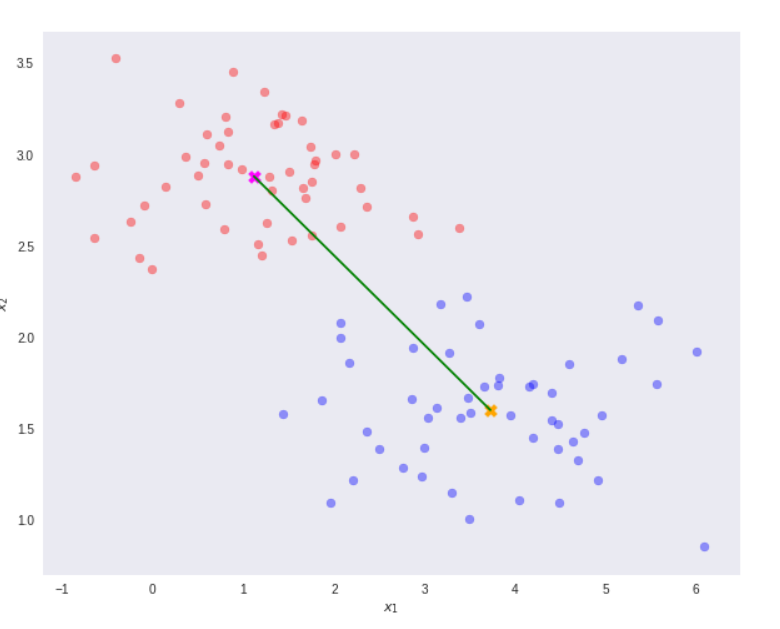
Take the dataset below as a toy example. We want to reduce the original data dimensions from D=2 to D’=1. In other words, we want a transformation T that maps vectors in 2D to 1D - T(v) = ℝ² →ℝ¹.

First, let’s compute the mean vectors **m1** and **m2** for the two classes.



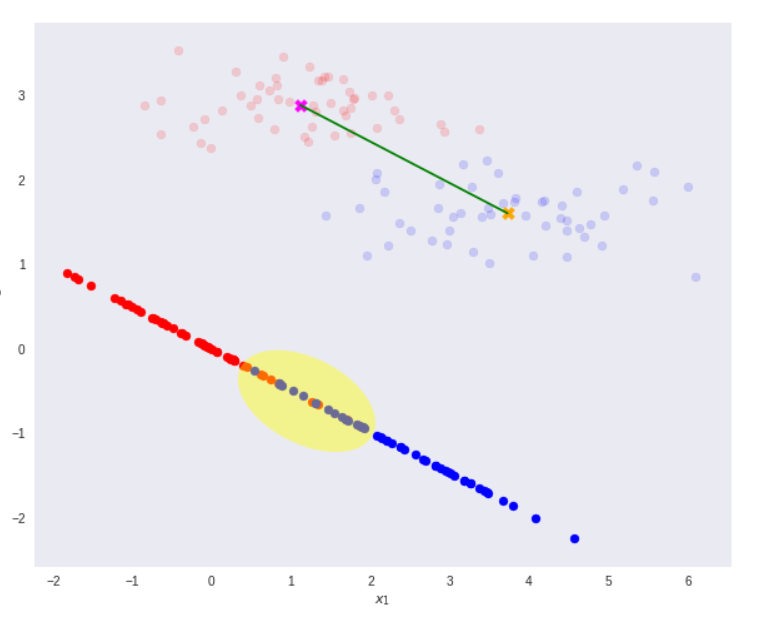


Note that N1 and N2 denote the number of points in classes C1 and C2 respectively. Now, consider using the class means as a measure of separation. In other words, we want to project the data onto the vector **W** joining the 2 class means.



It is important to note that any kind of projection to a smaller dimension might involve some loss of information. In this scenario, note that the two classes are clearly separable (by a line) in their original space.

However, after re-projection, the data exhibit some sort of class overlapping - shown by the yellow ellipse on the plot and the histogram below.





That is where the Fisher’s Linear Discriminant comes into play.

The idea proposed by Fisher is to maximize a function that will give a large separation between the projected class means while also giving a small variance within each class, thereby minimizing the class overlap.

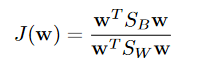
***In other words, FLD selects a projection that maximizes the class separation.*** To do that, it maximizes the ratio between the between-class variance to the within-class variance.

In short, to project the data to a smaller dimension and to avoid class overlapping, FLD maintains 2 properties.

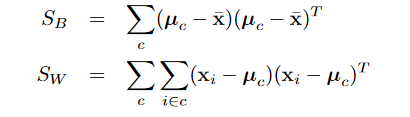
* A large variance among the dataset classes.
* A small variance within each of the dataset classes.

The most famous example of dimensionality reduction is ”principal components analysis”. This technique searches for directions in the data that have largest variance and subsequently project the data onto it. In this way, we obtain a lower dimensional representation of the data, that removes some of the ”noisy” directions. There are many difficult issues with how many directions one needs to choose, but that is beyond the scope of this note. PCA is an unsupervised technique and as such does not include label information of the data. For instance, if we imagine 2 cigar like clusters in 2 dimensions, one cigar has y= 1and the other y=−1. The cigars are positioned in parallel and very closely together, such that the variance in the total data-set, ignoring the labels, is in the direction of the cigars. For classification, this would be a terrible projection, because all labels get evenly mixed and we destroy the useful information. A much more useful projection is orthogonal to the cigars, i.e. in the direction of least overall variance, which would perfectly separate the data-cases (obviously, we would still need to perform classification in this 1-D space).

So the question is, how do we utilize the label information in finding informative projections? To that purpose Fisher-LDA considers maximizing the following objective:



Where SB is the “between classes scatter matrix” and SW is the “within classes scatter matrix”. Note that due to the fact that scatter matrices are proportional to the covariance matrices we could have defined. Jusing covariance matrices – the proportionality constant would have no effect on the solution. The definitions of the scatter matrices are:



where ̄x is the overall mean of the data-cases. Oftentimes you will see that for 2 classes SB is defined as S′B= (μ1−μ2)(μ1−μ2)T. This is the scatter of class 1 with with respect to+

the scatter of class 2 and hence corresponds to computing the scatter relative to a different +vector. By using the general transformation rule for scatter matrices:



With we can deduce that the only difference is a constant shift not depending on any relative distances between point. It will therefore have no impact on the final solution.

The links used are

<https://sthalles.github.io/fisher-linear-discriminant/>

<https://www.ics.uci.edu/~welling/teaching/273ASpring09/Fisher-LDA.pdf>

<https://www.sciencedirect.com/topics/engineering/fisher-linear-discriminant>

<https://cedar.buffalo.edu/~srihari/CSE555/Chap3.Part6.pdf>