Quantitative Text Analysis

Meeting 7

Petro Tolochko

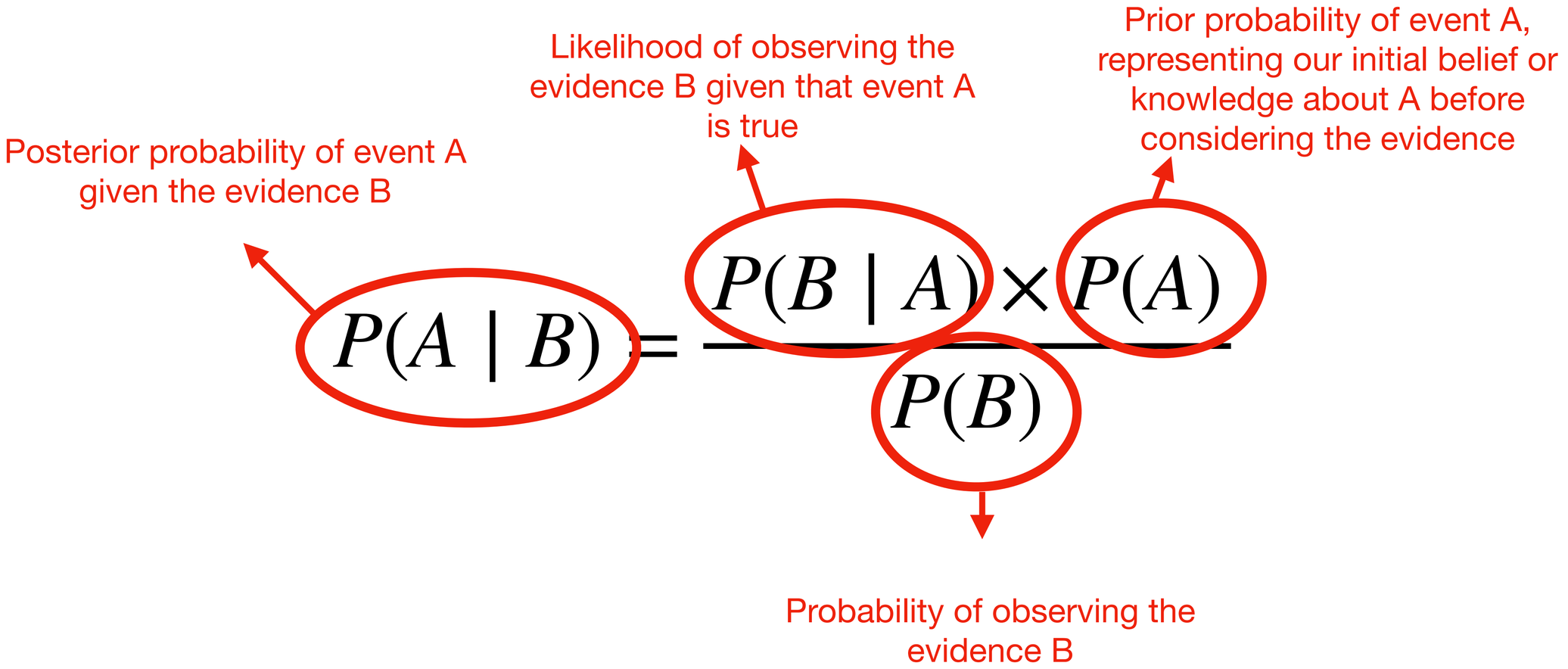
Naïve Bayes Classifier

- Probabilistic classifier
- Simple
- Fast
- Good Accuracy

Bayes Theorem

$$P(A \mid B) = \frac{P(B \mid A) \times P(A)}{P(B)}$$

Bayes Theorem



Bayes Theorem

$$P(A|B) \propto P(B|A) \times P(A)$$

Naïve Bayes Classifier

$$P(C_k|x_1, x_2, ..., x_n)$$

Naïve Bayes Classifier

Features

$$P(C_k|x_1, x_2, ..., x_n)$$

Class

Naïve Bayes Classifier

Features

$$P(C_k|x_1, x_2, ..., x_n)$$

Class

Features are assumed to be independent. Hence, "***Naïve***"

Naïve Bayes Classifier

$$P(C_k|\mathbf{x}) = \frac{P(C_k) \times P(\mathbf{x}|C_k)}{P(\mathbf{x})}$$

Naïve Bayes Classifier

$$P(C_k|\mathbf{x}) \propto P(C_k) \times P(\mathbf{x}|C_k)$$

Naïve Bayes Classifier

$$
\begin{aligned}
p(C_k \mid x_1, \ldots, x_n) &\propto p(C_k, x_1, \ldots, x_n) \\
&\propto p(C_k)\, p(x_1 \mid C_k)\, p(x_2 \mid C_k)\, p(x_3 \mid C_k)\, \cdots \\
&\propto p(C_k) \prod_{i=1}^{n} p(x_i \mid C_k)\,,
\end{aligned}
$$

Decision Rule

$$\hat{y} = argmax\ p(C_k) \prod_{i=1}^{n} p(x_i|C_k)$$

Naïve Bayes Classifier

- Implemented in many stats/ML packages

Support Vector Machine

- Comes from computer science
- Very good
- Rather difficult math

- Considered one of the best of-the-shelf classification algorithms

Hyperplane

- n-1 dimensional plane that separates the n-dimensional space

Hyperplane

- n-1 dimensional plane that separates the n-dimensional space
- 2-dimensional hyperplane:

$$\beta_0 + \beta_1 X_1 = 0$$

Hyperplane

- n-1 dimensional plane that separates the n-dimensional space
- 2-dimensional hyperplane:
- line equation

$$\beta_0 + \beta_1 X_1 = 0$$

Hyperplane

- n-1 dimensional plane that separates the n-dimensional space
- 2-dimensional hyperplane:
- line equation

$$\beta_0 + \beta_1 X_1 = 0$$

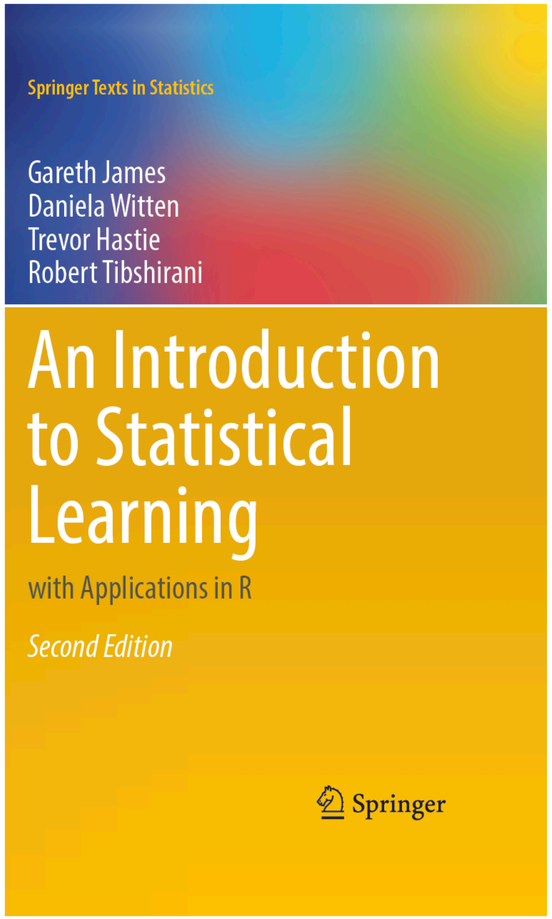
$$\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_p X_p = 0$$

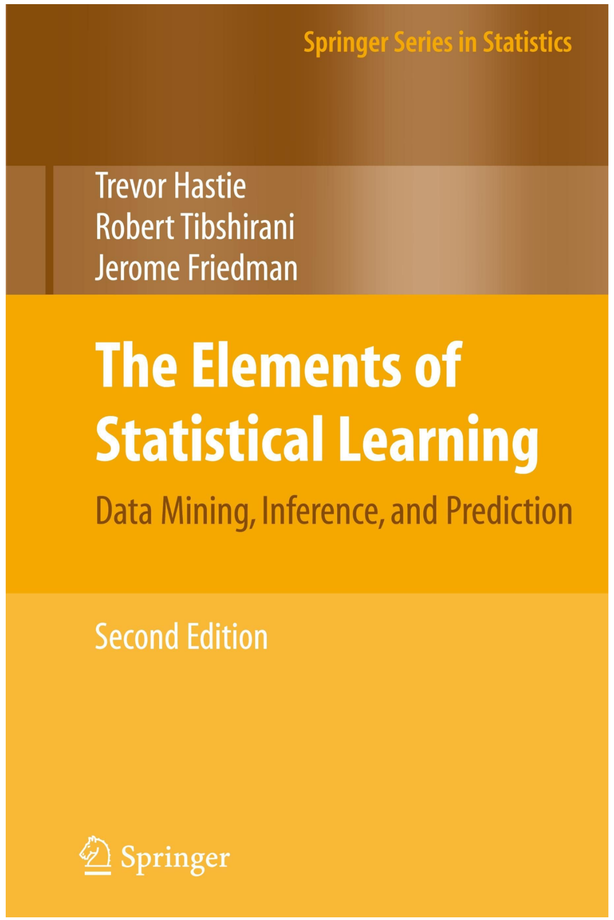
Classification

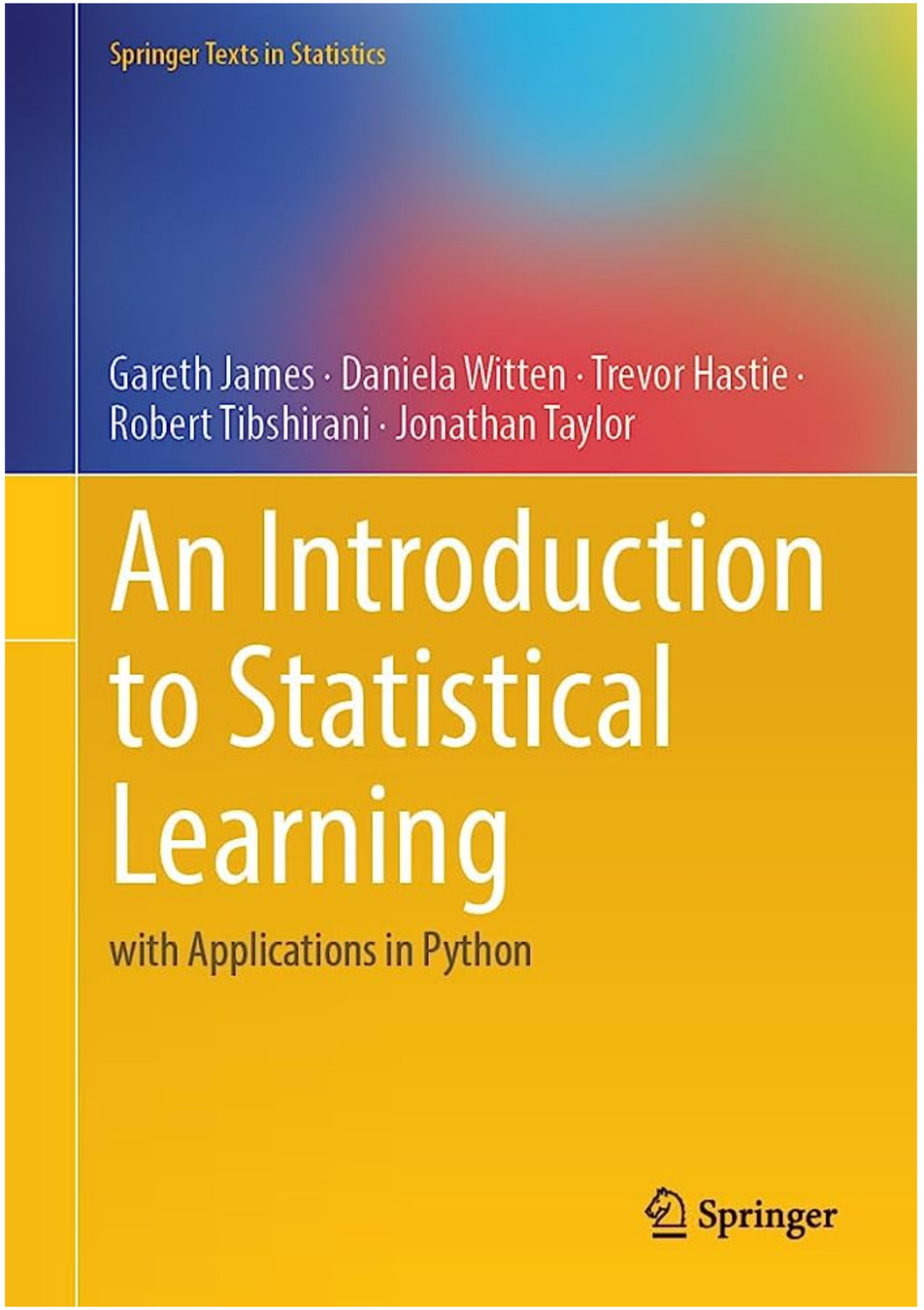
$$\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_p X_p > 0.$$

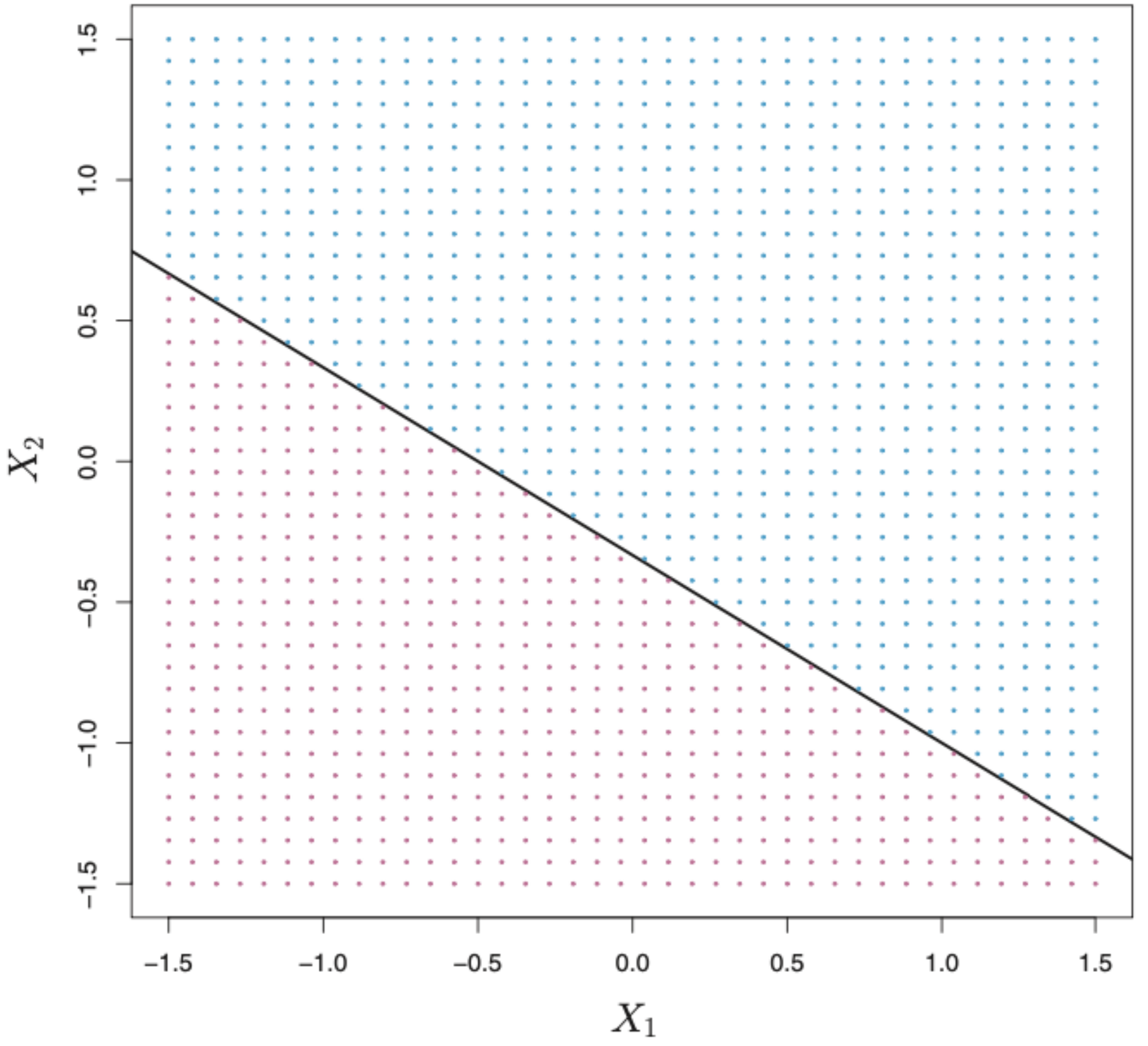
$$\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_p X_p < 0$$

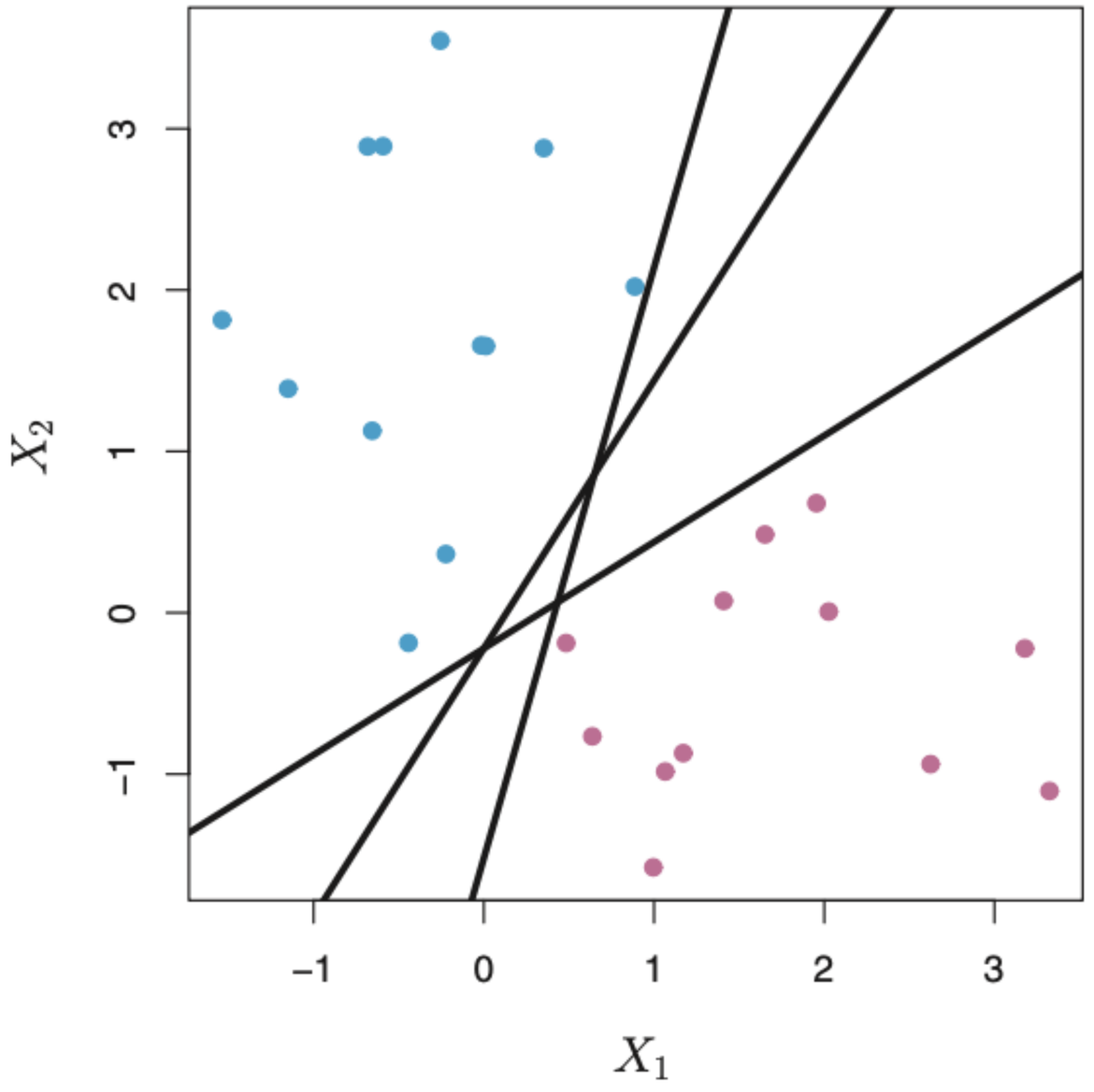
Following images from:

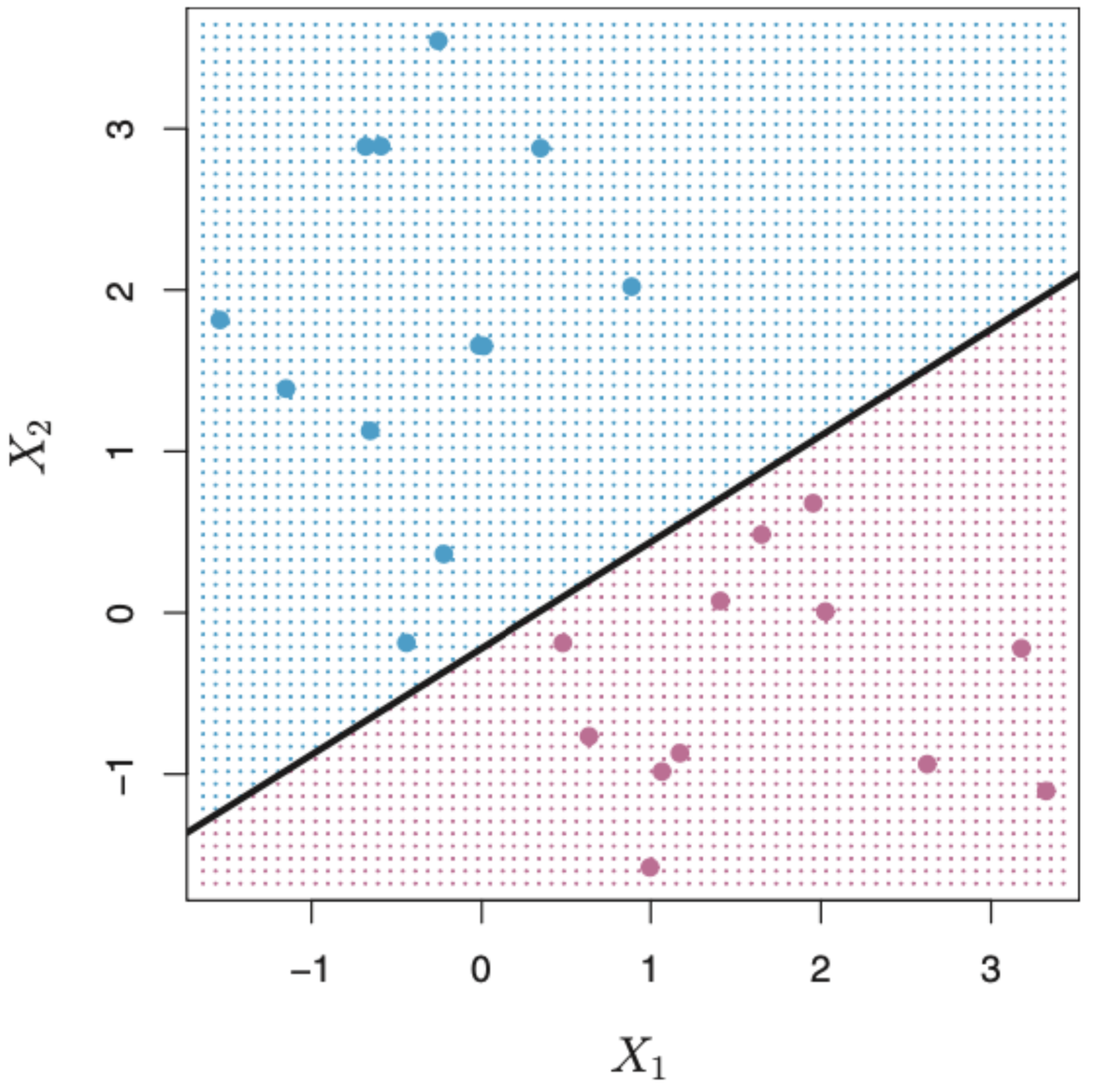




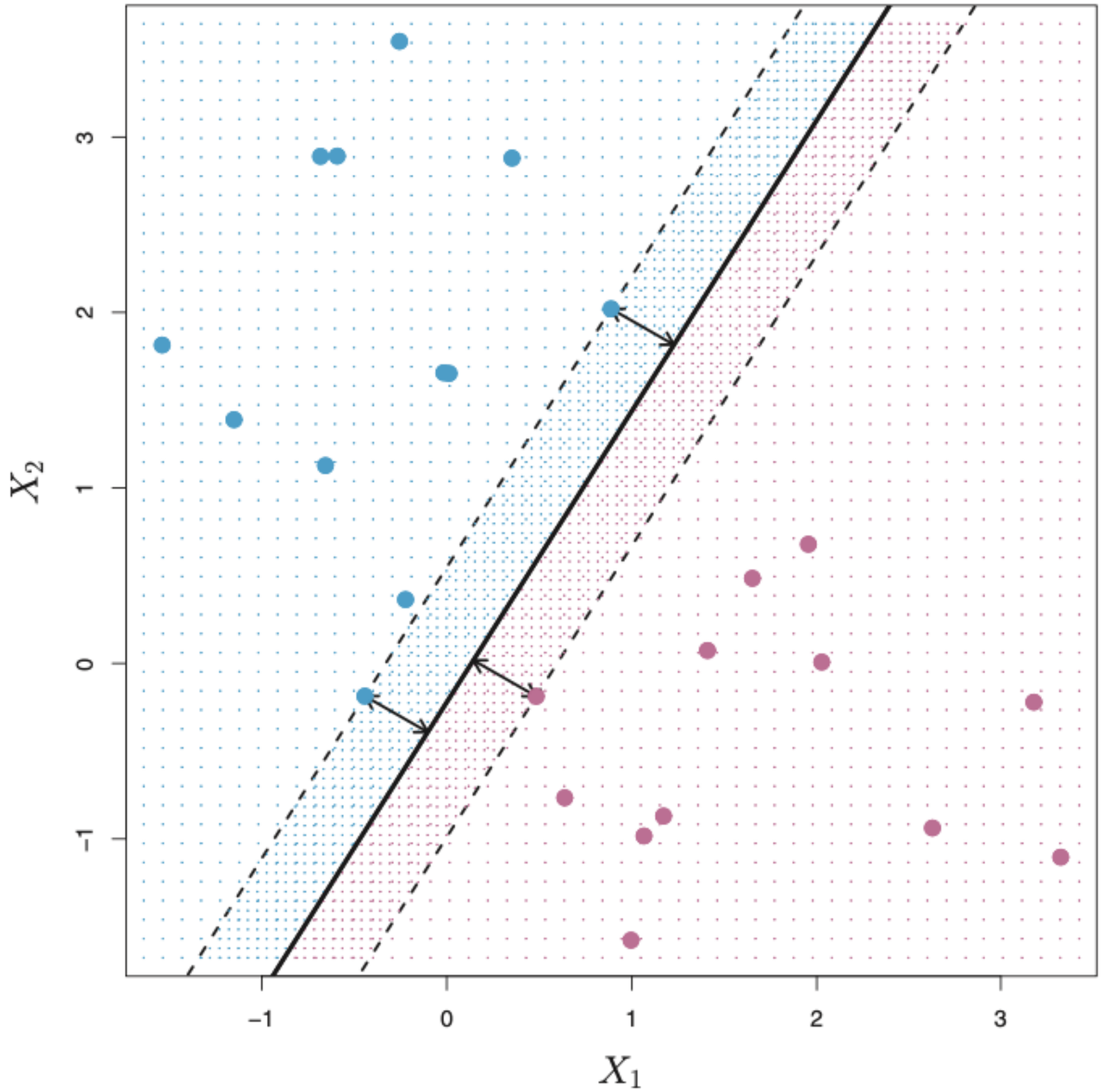

Springer Texts in Statistics
Gareth James · Daniela Witten · Trevor Hastie ·
Robert Tibshirani · Jonathan Taylor
An Introduction to Statistical Learning
with Applications in Python
Springer


1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
X_2
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
X_1


X_2
3
2
1
0
−1
−1
0
1
2
3
X_1


X_2
3
2
1
0
−1
−1
0
1
2
3
X_1

SV Classifier



Support Vector Machine

- Non-linear version of the Support Vector Classifier
- Extension using Kernels

Support Vector Machines

$$f(x) = \beta_0 + \sum_{i \in \mathcal{S}} \alpha_i \langle x, x_i \rangle$$

$$f(x) = \beta_0 + \sum_{i \in \mathcal{S}} \alpha_i K(x, x_i)$$

Support Vector Machines

$$f(x) = \beta_0 + \sum_{i \in \mathcal{S}} \alpha_i \langle x, x_i \rangle$$

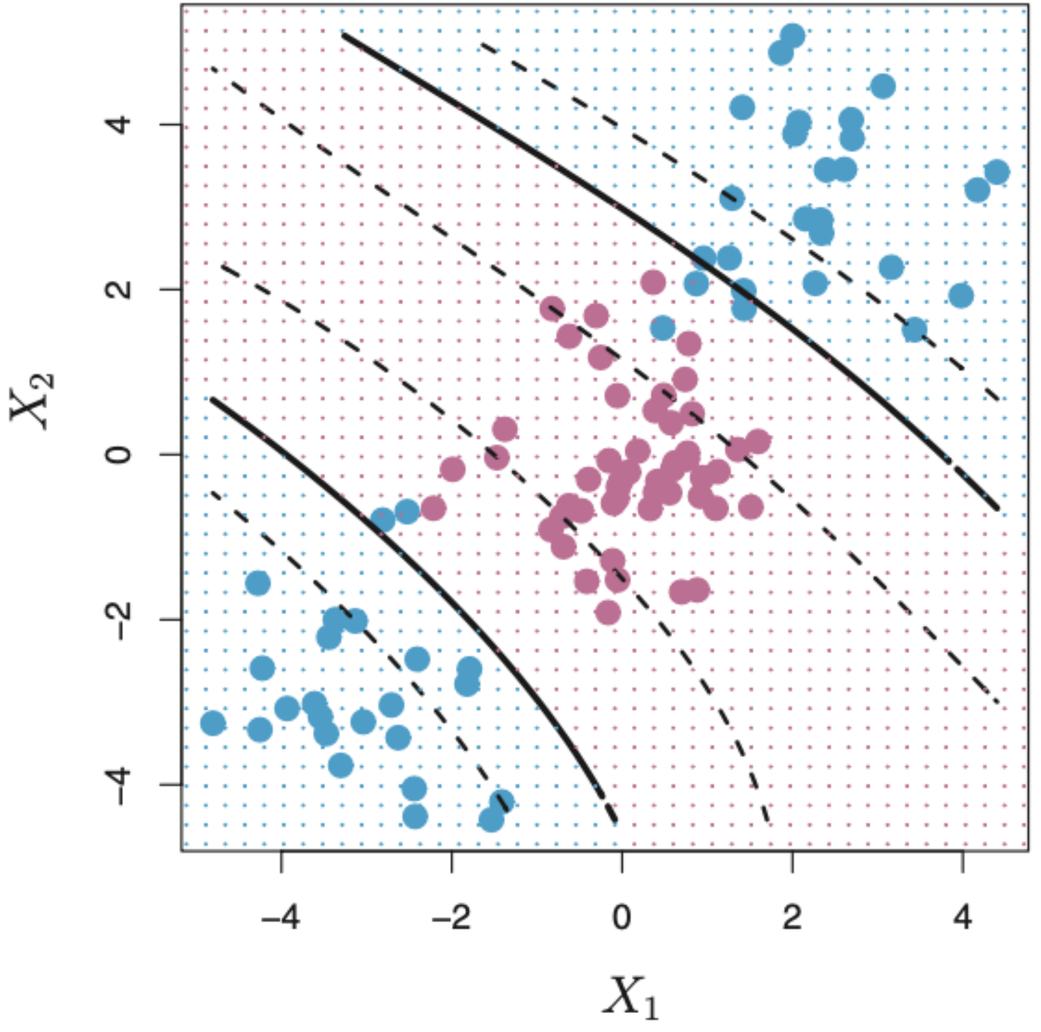
$$f(x) = \beta_0 + \sum_{i \in \mathcal{S}} \alpha_i K(x, x_i)$$

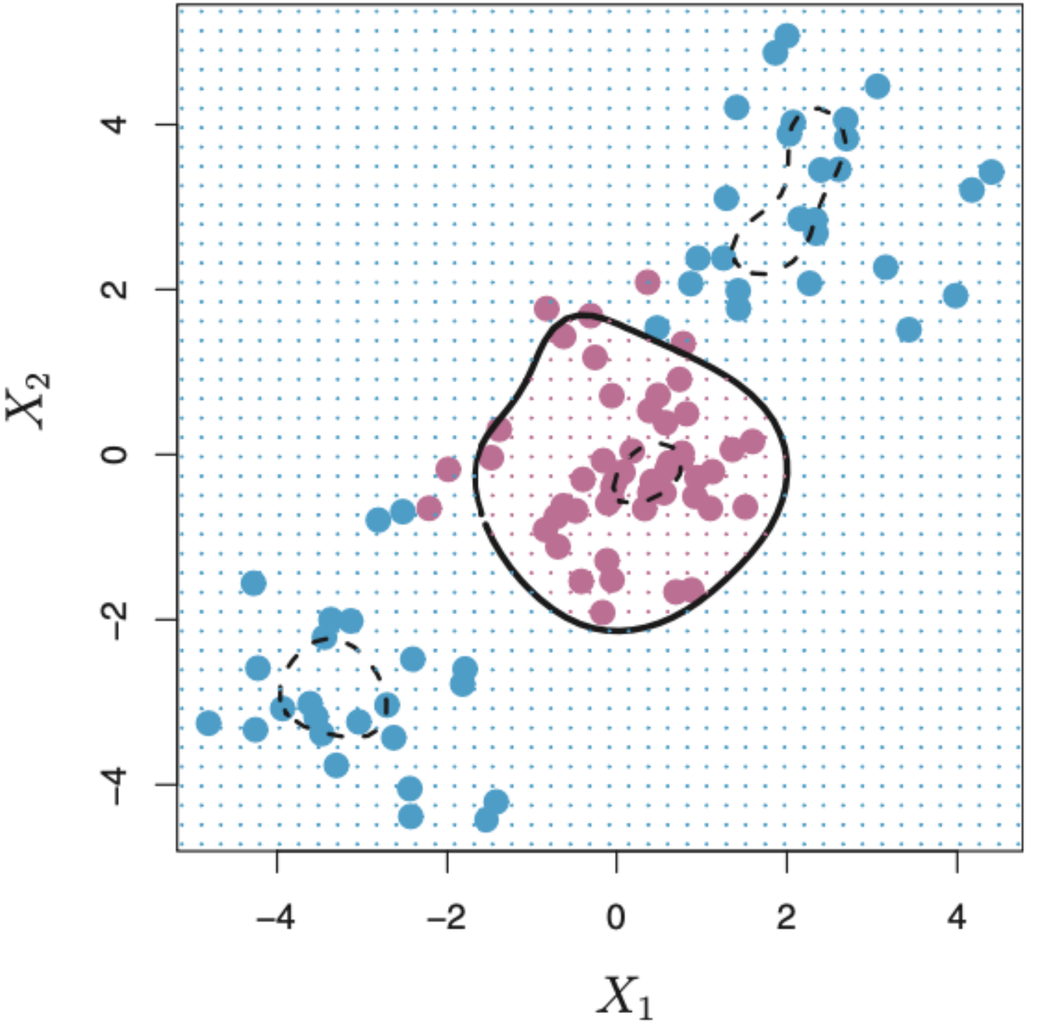
Kernel function

Support Vector Machines

$$K(x_i, x_{i'}) = (1 + \sum_{j=1}^{p} x_{ij} x_{i'j})^d$$

Polynomial Kernel

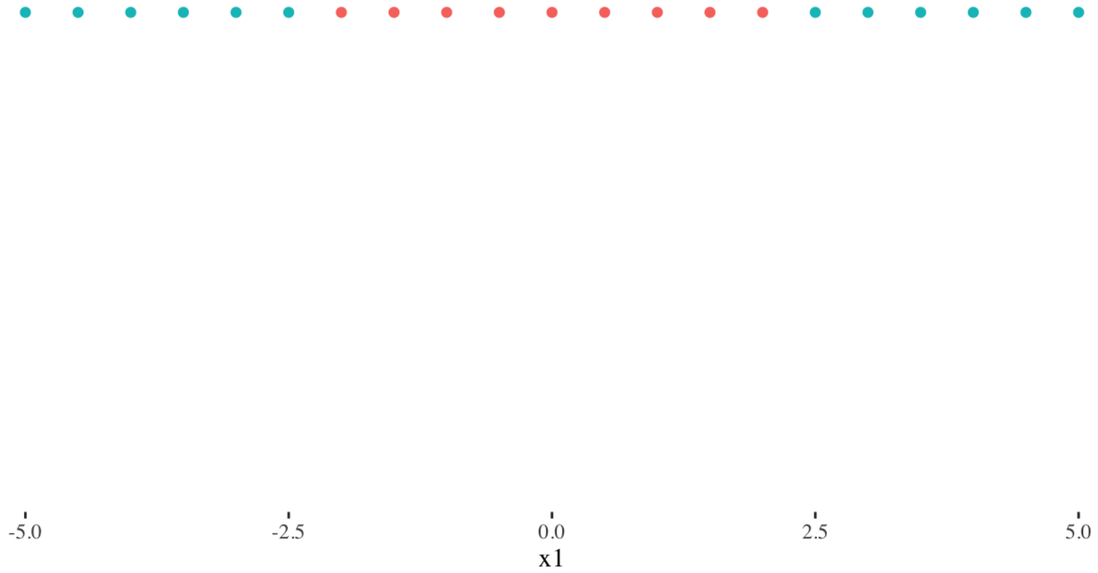

X_2
4
2
0
−2
−4
−4
−2
0
2
4
X_1

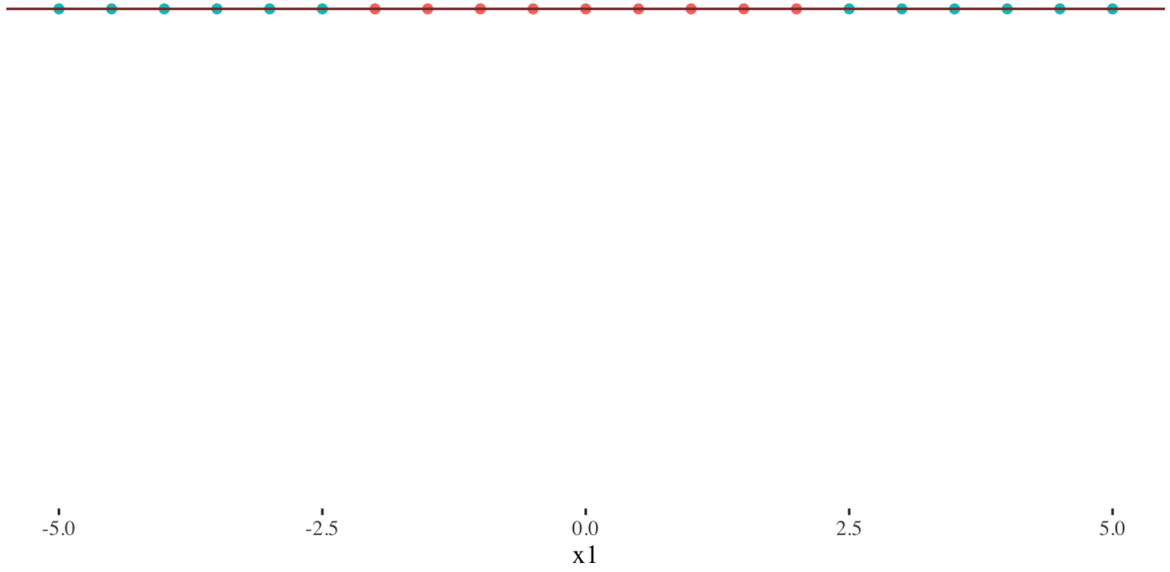

X_2
4
2
0
−2
−4
−4
−2
0
2
4
X_1

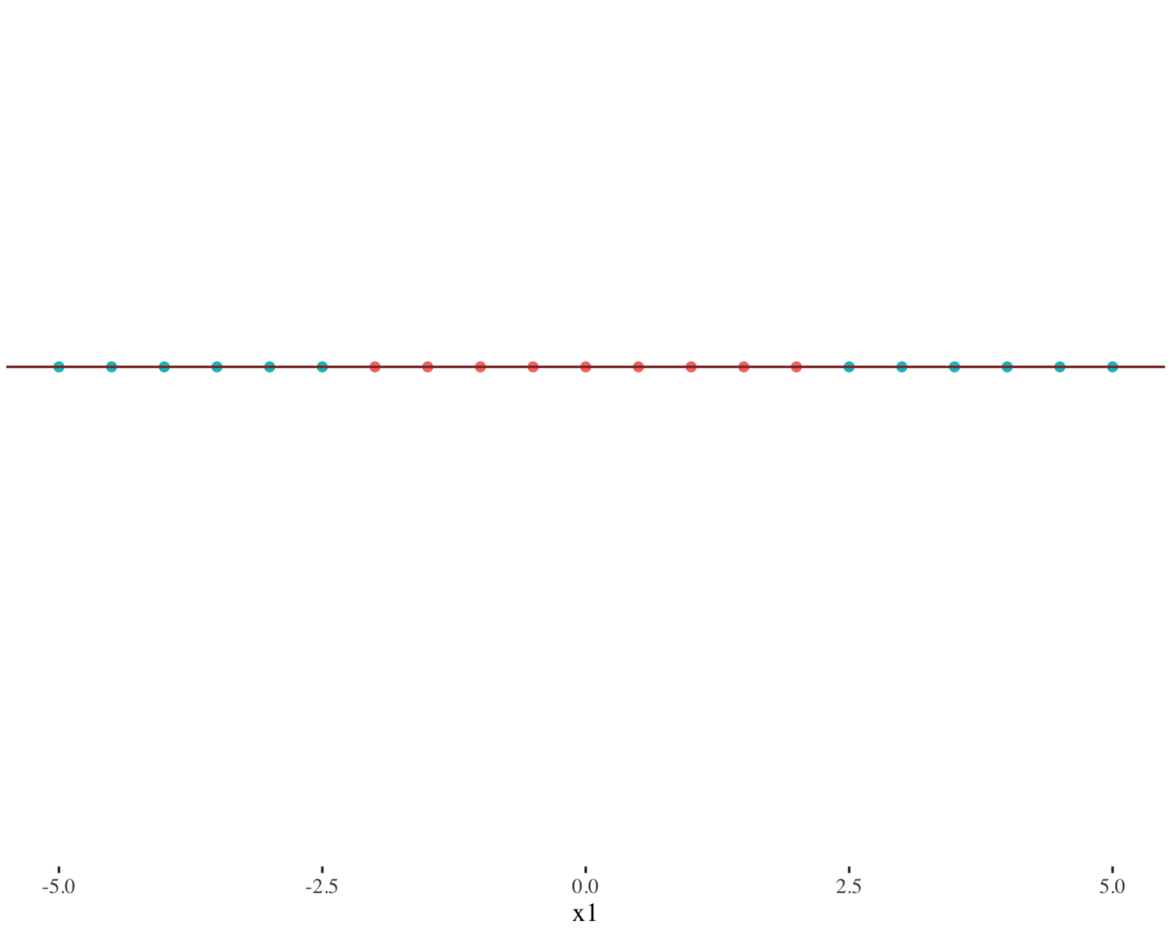
Kernel Trick

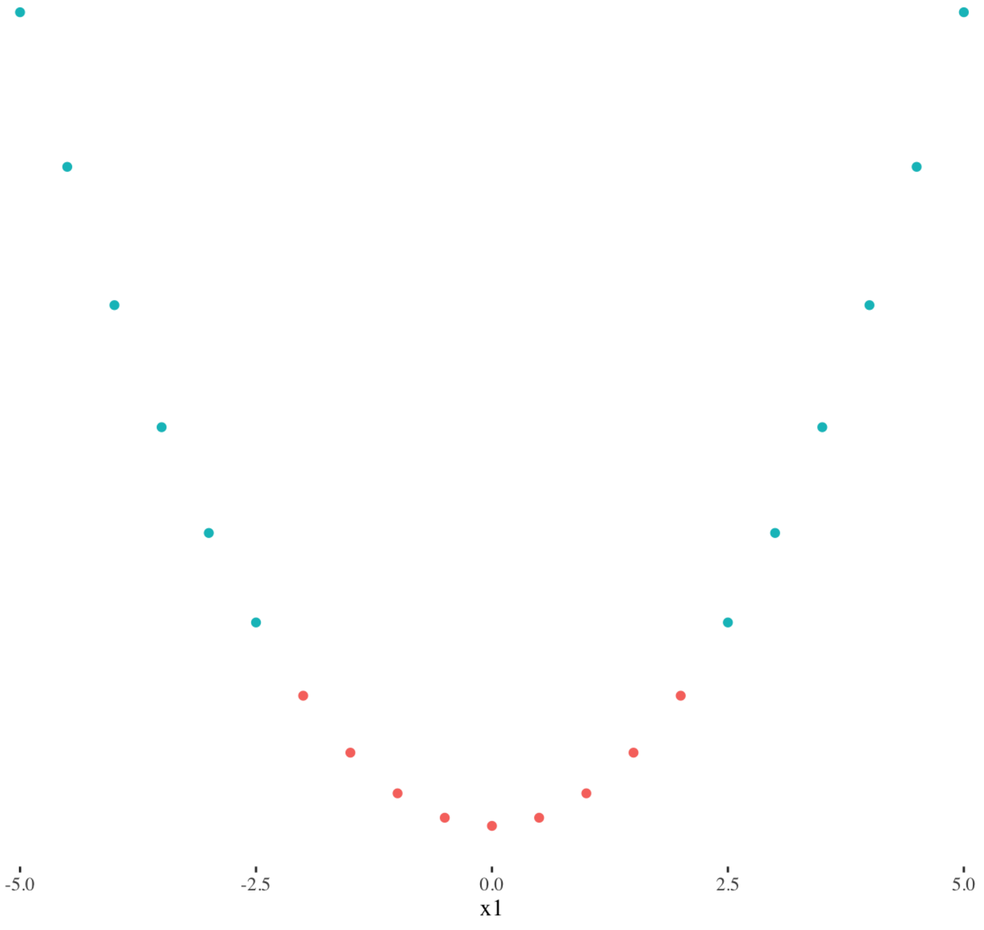
Kernel Trick

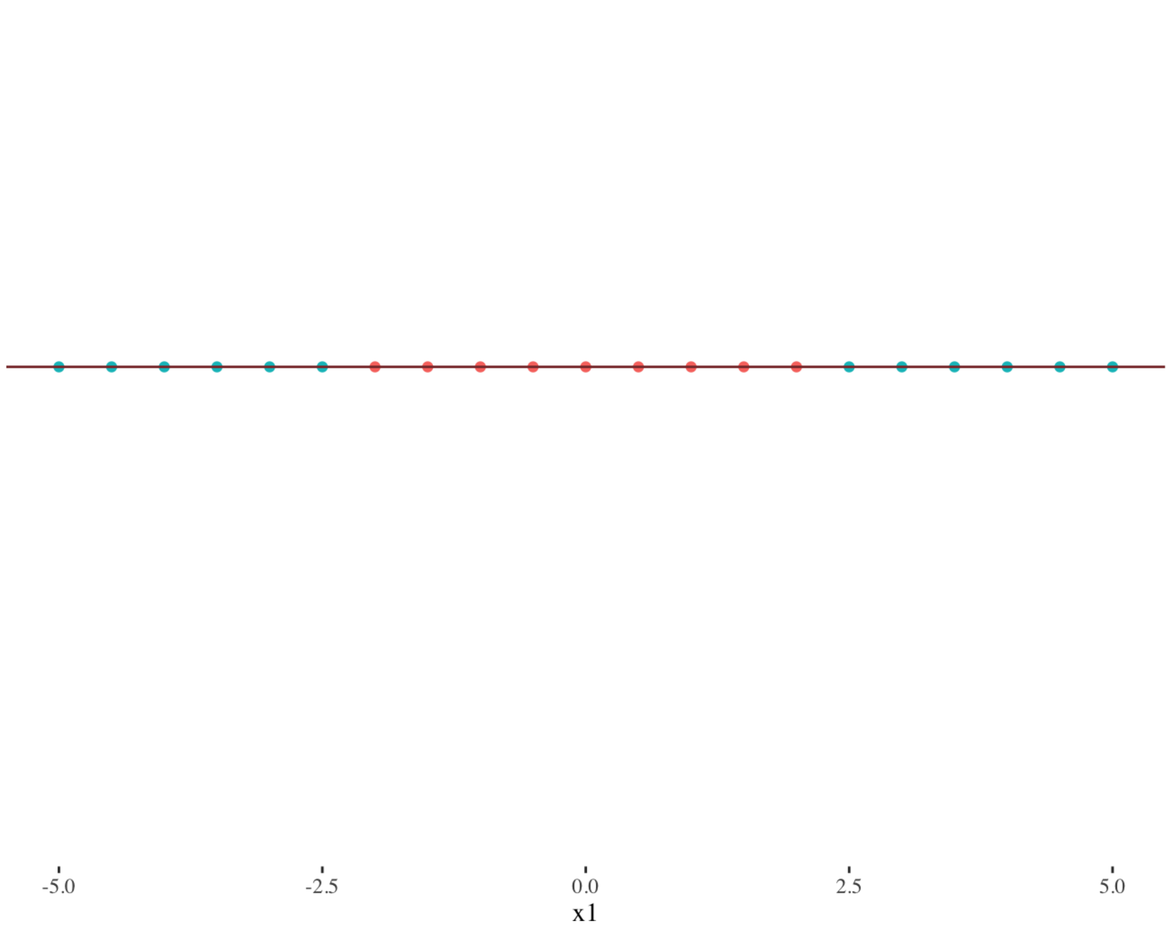
- Actual name
- Attempt to place n-dimensional data into n+1 dimensional space

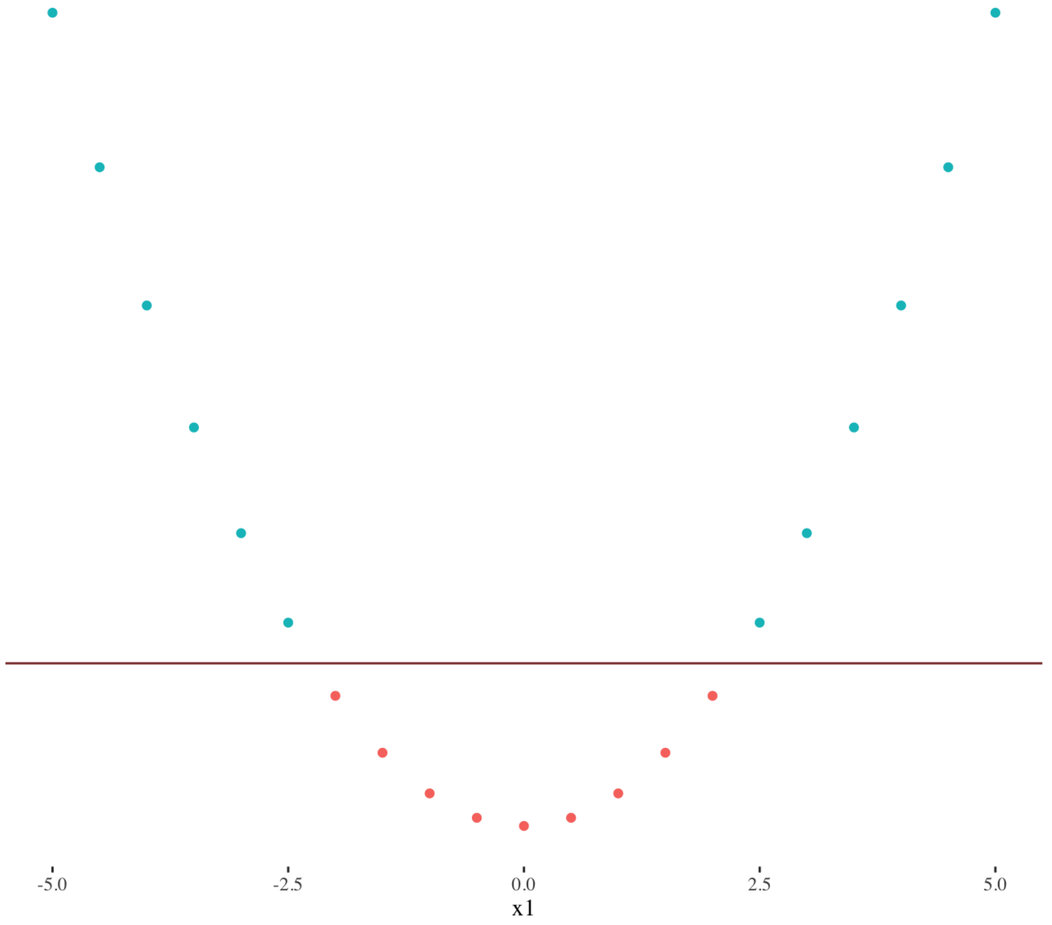

-5.0
-2.5
0.0
2.5
5.0
x1

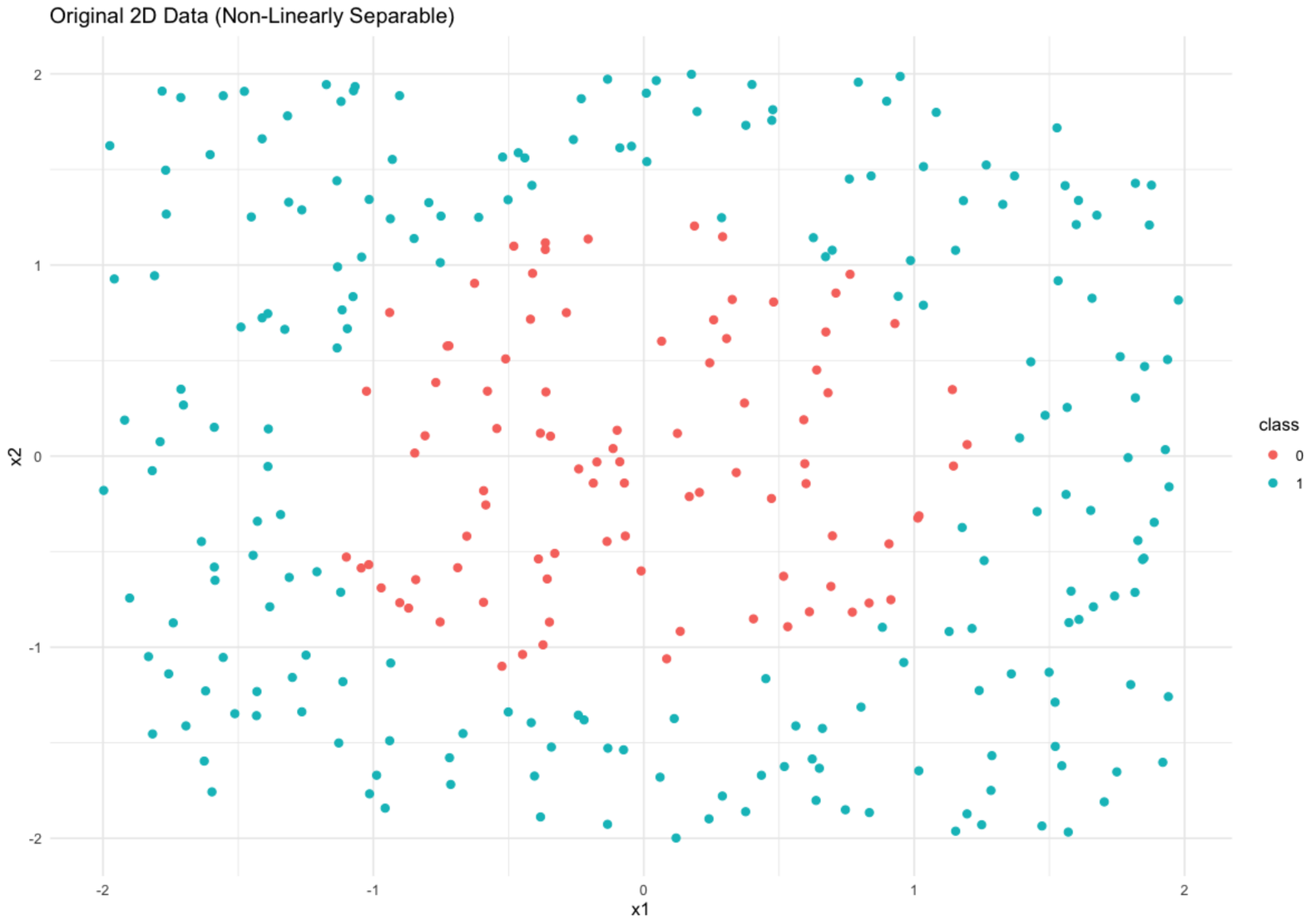

-5.0
-2.5
0.0
2.5
5.0
x1

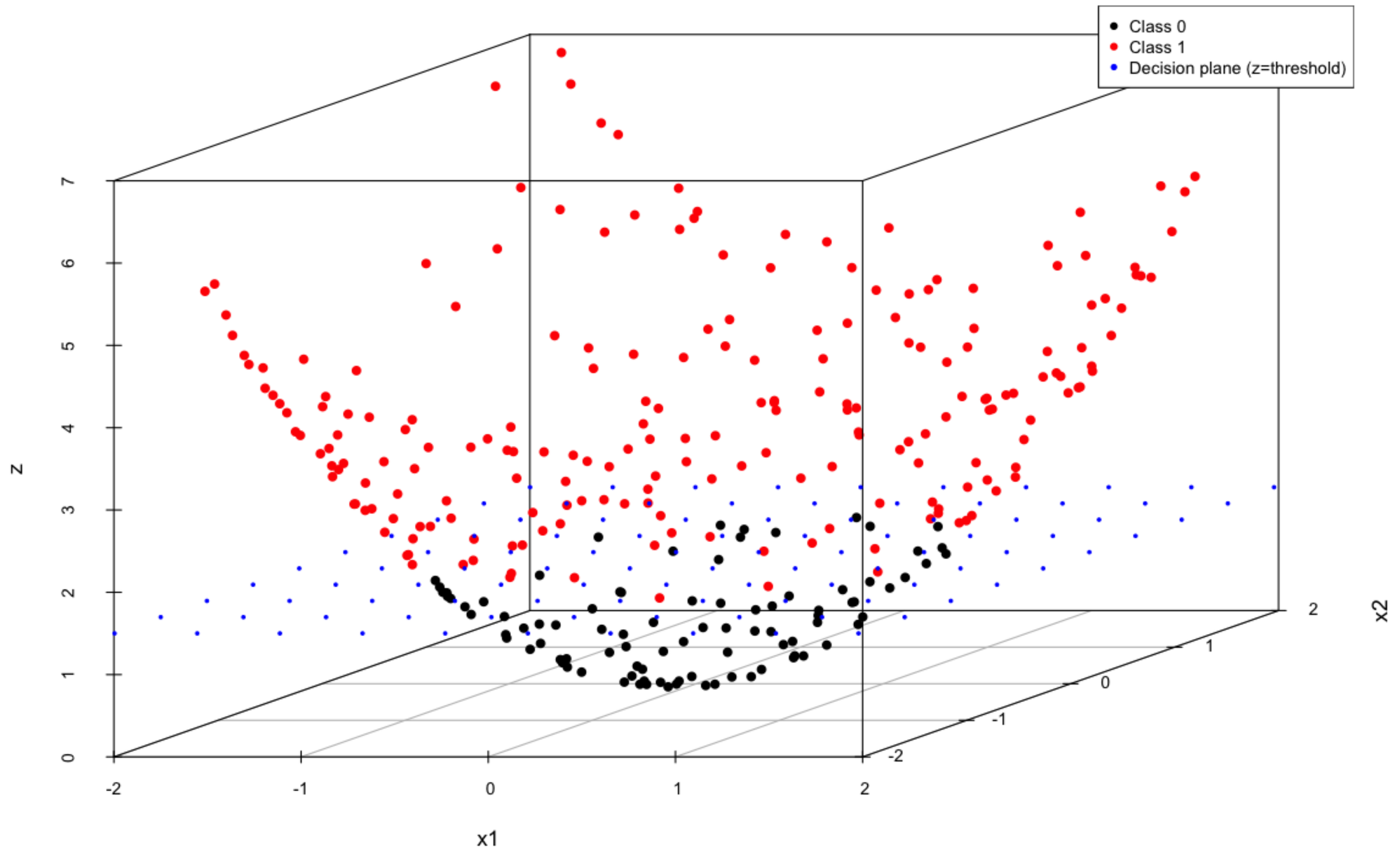

-5.0
-2.5
0.0
2.5
5.0
x1

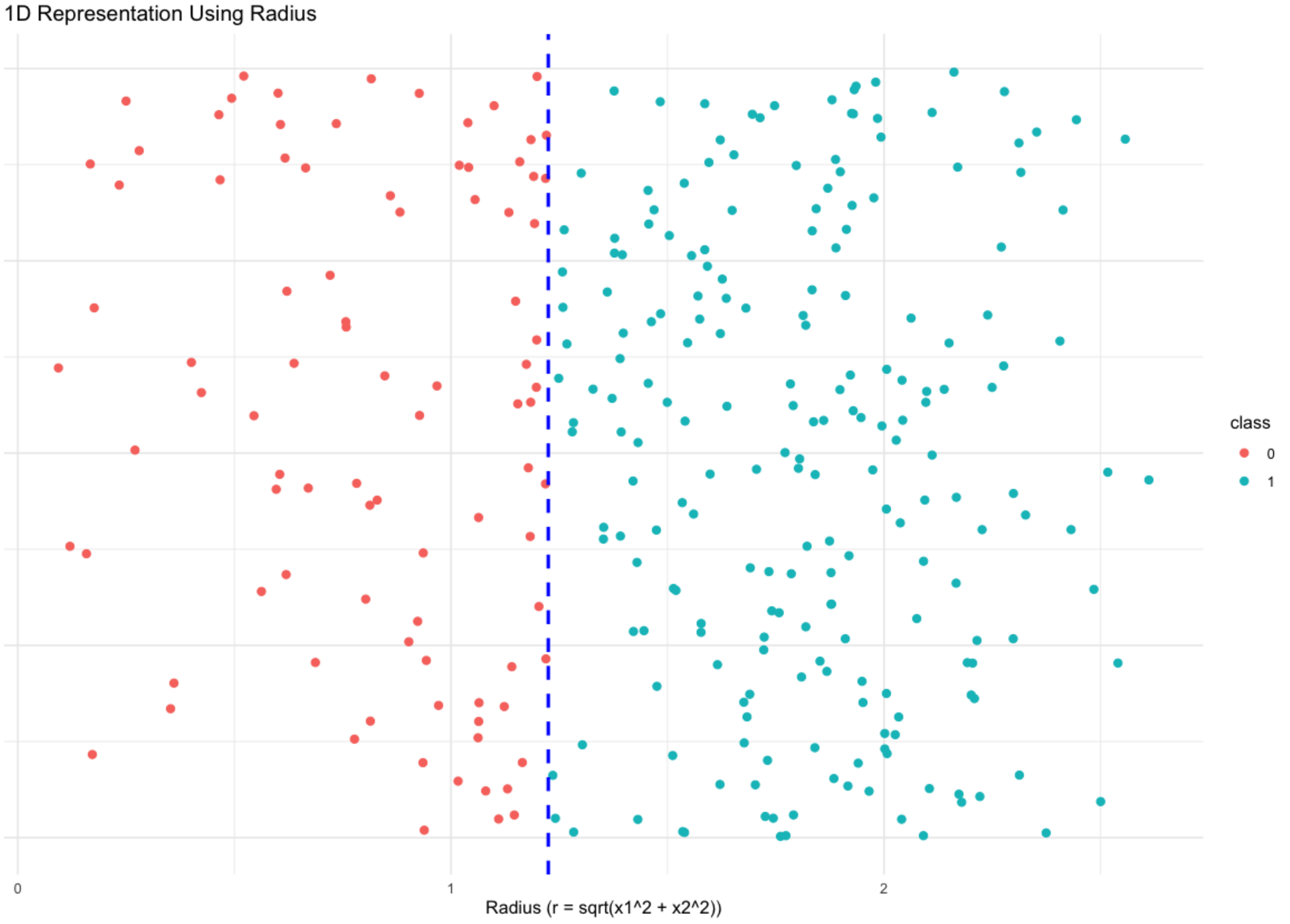

-5.0
-2.5
0.0
2.5
5.0
x1


-5.0
-2.5
0.0
2.5
5.0
x1


-5.0
-2.5
0.0
2.5
5.0
x1


Original 2D Data (Non-Linearly Separable)
x2
x1
2
1
0
-1
-2
-2
-1
0
1
2
class
0
1


3D Transformed Data
(x1, x2, z = x1^2+x2^2)
Class 0
Class 1
Decision plane (z=threshold)
z
0
1
2
3
4
5
6
7
x1
-2
-1
0
1
2
x2
-2
-1
0
1
2


1D Representation Using Radius
class
0
1
0
1
2
Radius (r = sqrt(x1^2 + x2^2))