University of Moratuwa
Department Electronics and Telecommunication

Assignment 04 : Kernel Methods

Senior Lecturer
Dr. Sampath Perera

Candidate
K.Garthigan - 200183N

Year 2023

Abstract

This report elucidates solutions to EN3150 Module Assignment 02, emphasizing the application of kernel methods for the Support Vector Machine (SVM) algorithm in handling non-separable data. Through clear and concise explanations, accompanied by code snippets and results, the document navigates the intricacies of these advanced techniques. As we mark the one-year milestone since its inception, this report remains a beacon of clarity, offering a succinct and accessible guide to harnessing kernel methods for enhanced SVM performance in challenging data scenarios.

1. Kernel Methods

a. Question 1

$$\phi(x) = (x, \sqrt{2}x\ , x^2)$$

for one dimensional data

$$\phi(z) = \left(z, \sqrt{2}z\ , z^2\right)$$

$$\phi(z) * \phi(x) = (1 + 2zx + x^2z^2)$$

$$K(x,z) = (1 + 2xz + z^2x^2)$$

$$K(x,z) = (1 + xz)^2$$

b. Question 2

$$K(x,z) = (1 + xz)^2$$

For two dimensional data

$$K(x,z) = (1 + x_1z_1 + x_2z_2)^2$$

c. Question 3

$$K(x,z) = (1 + x_1z_1 + x_2z_2)^2$$

$$K(x,z) = (1 + (x_1z_1)^2 + (x_2z_2)^2 + 2x_1z_1 + \ 2x_2z_2 + \ 2x_1z_1x_2z_2$$

d. Question 4

$$G = \begin{bmatrix} k(x_1,x_1) & k(x_1,x_2) & k(x_1,x_3) & k(x_1,x_4) \\ k(x_2,x_1) & k(x_2,x_2) & k(x_2,x_3) & k(x_2,x_4) \\ k(x_3,x_1) & k(x_3,x_2) & k(x_3,x_3) & k(x_3,x_4) \\ k(x_4,x_1) & k(x_4,x_2) & k(x_4,x_3) & k(x_4,x_4) \end{bmatrix}$$

$$G = \begin{bmatrix} 27^2 & 24^2 & 15^2 & 71^2 \\ 24^2 & 26^2 & 21^2 & 79^2 \\ 15^2 & 21^2 & 21^2 & 65^2 \\ 71^2 & 79^2 & 65^2 & 245^2 \end{bmatrix}$$

$$G = \begin{bmatrix} 729 & 576 & 225 & 5041 \\ 576 & 676 & 441 & 6241 \\ 225 & 441 & 441 & 4225 \\ 5041 & 6241 & 4225 & 60025 \end{bmatrix}$$

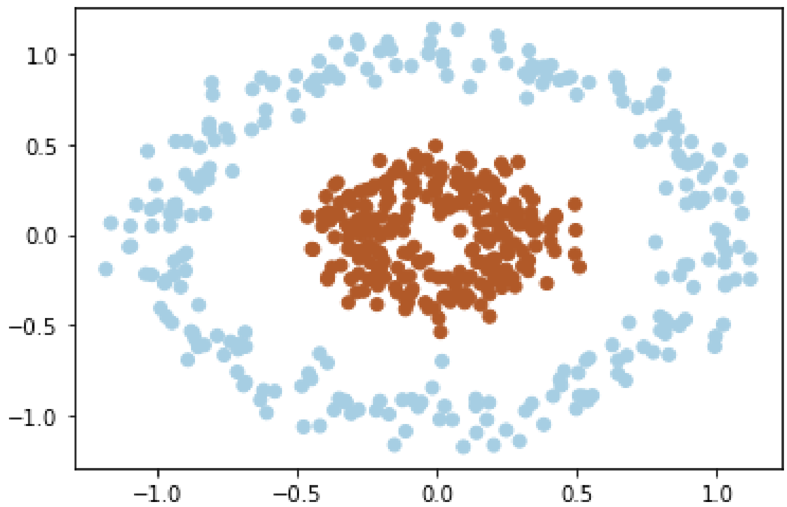
For this 4*4 matrix eigen values are

315.858 , 2.8946 , 0.0096 , 9.2378

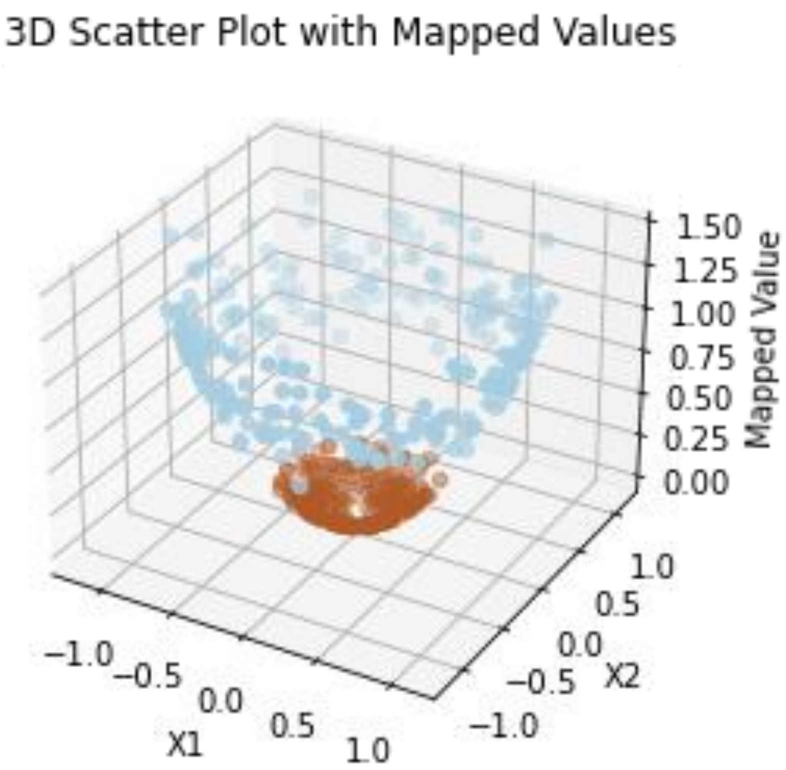
If all the eigenvalues of a matrix are non-negative, and the matrix is symmetric, then it qualifies as a positive semi-definite matrix. This property also implies that the matrix can serve as a valid kernel.

2. Results

a.

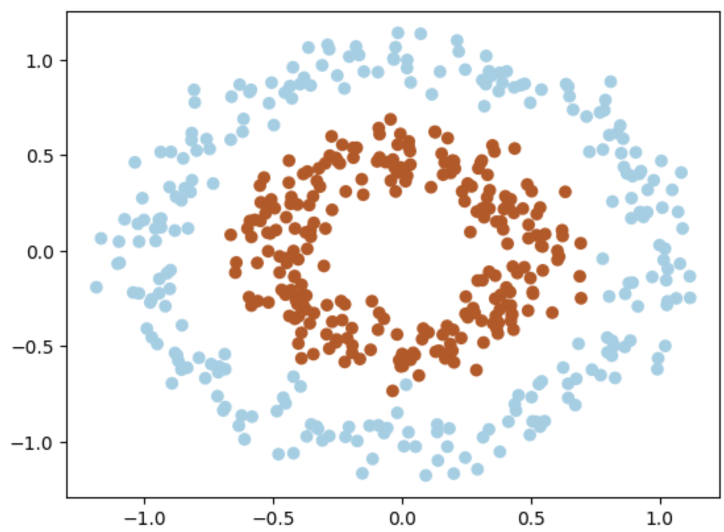


When we are using mapping function to visualize in 3D

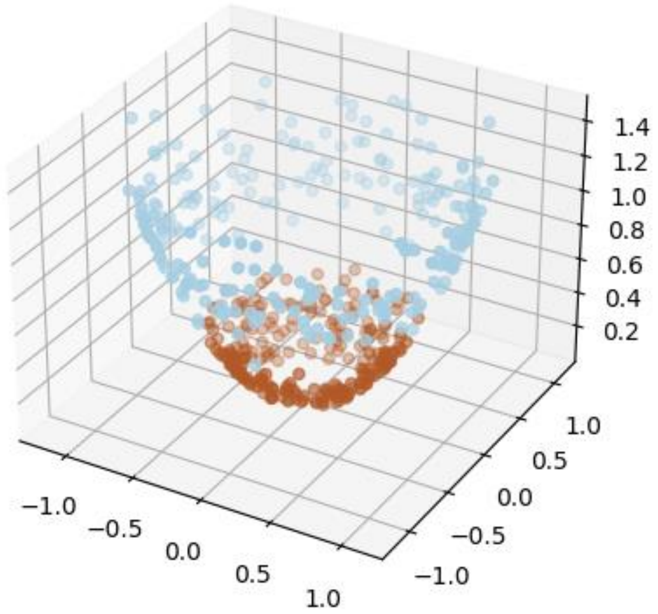


We generated a dataset and introduced a factor of 0.3, then applied the previously mentioned function to project the data into a higher-dimensional space. Upon visualizing the results, it became apparent that the data exhibits linear separability. We've effectively transformed the data to a higher-dimensional representation, making it suitable for training a linear support vector machine (SVM) algorithm.

When if we are increasing the factor from 0.3 to 0.5

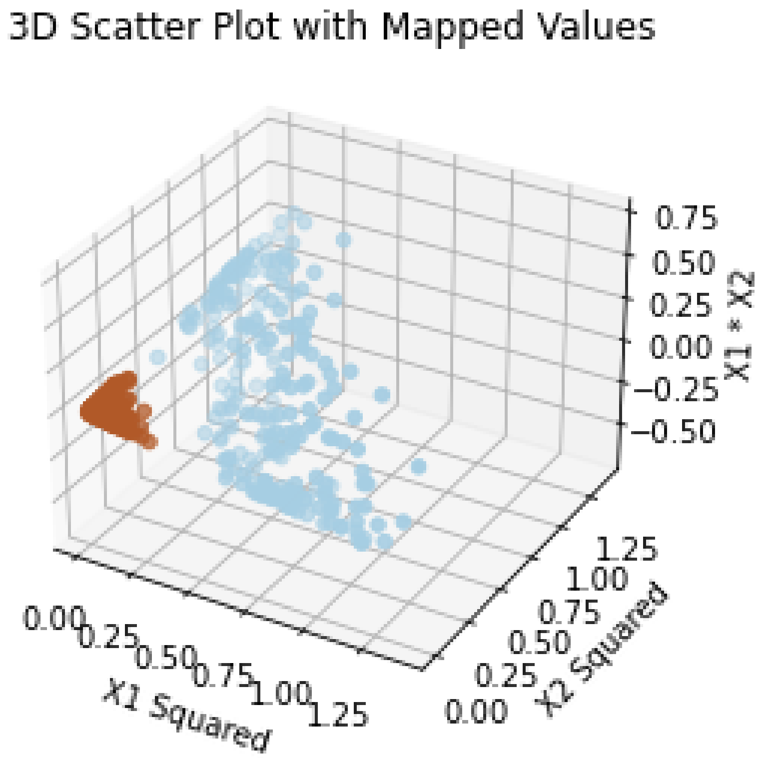


When mapping above one



Here the data from the two classes tries to mix up so 0.3 is the optimal one to use it as a radius.

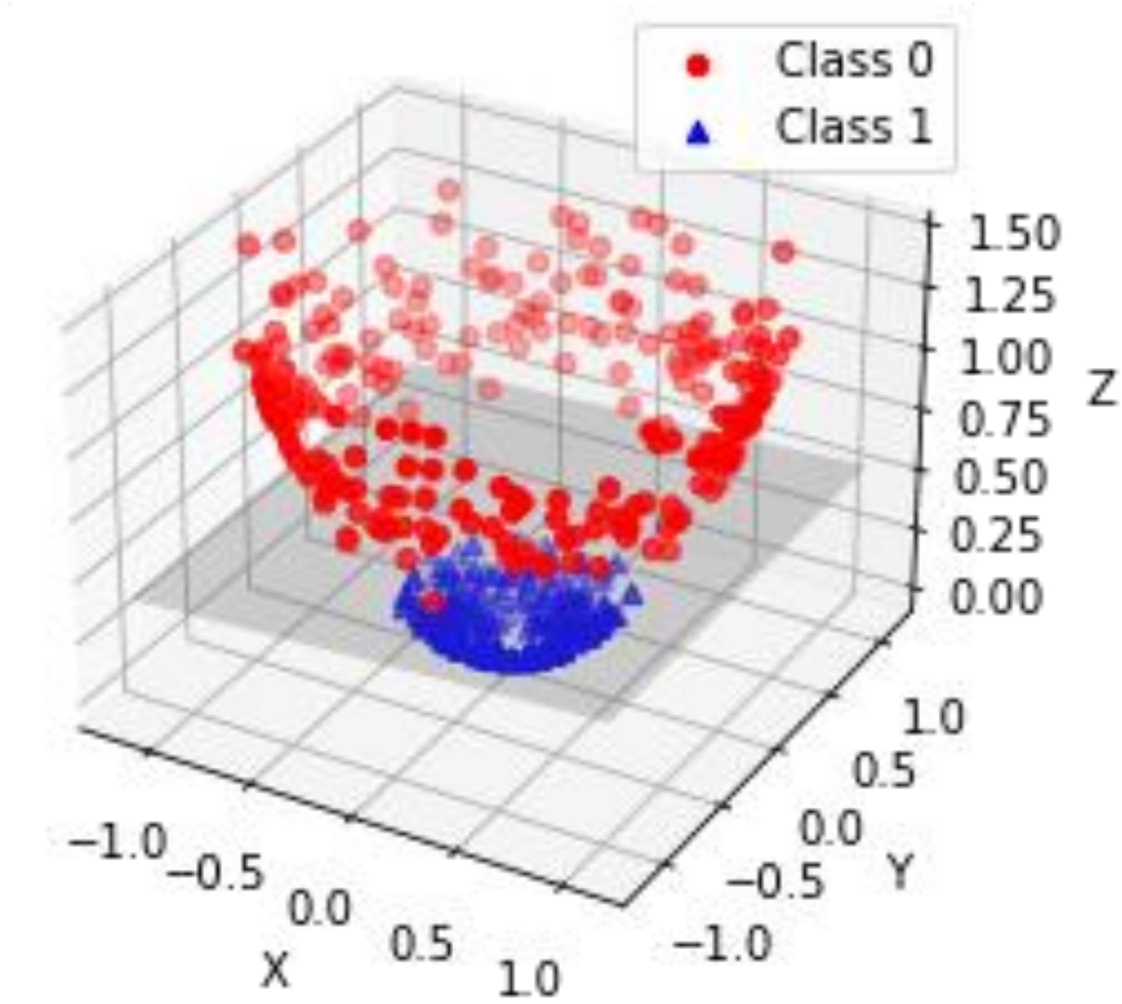
When we change the mapping function
As $\varphi : x = (x_1,x_2) \rightarrow \varphi(x) = (x_1^2,x_2^2,x_1x_2)$



the mapping we're referring to involves squaring the features (x1 and x2) to create a higher-dimensional space. By doing this nonlinear transformation, the data now shows distinct clusters in this expanded space. This transformation makes it easier to draw a straight line (linear decision boundary) to separate these clusters effectively. So, even though the original data might not have been easily separable with a linear boundary, the squared features create a more structured and distinguishable pattern, allowing for a simpler linear separation.

Running Linear SVC

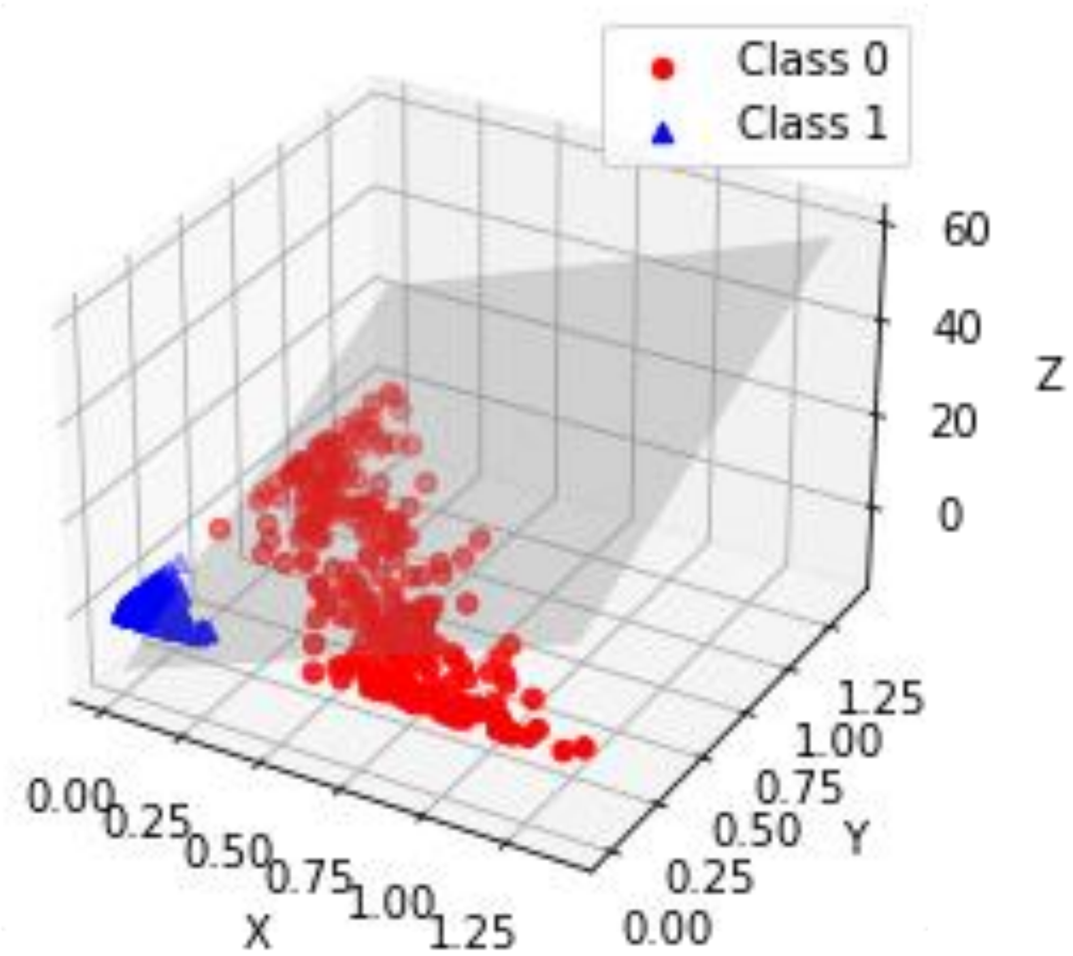
For data with first mapping



Results were

Accuracy = 1.0
Recall = 1.0
Precision = 1.0
F1 Score = 1.0

For data with second mapping



Results were

Accuracy = 1.0
Recall = 1.0
Precision = 1.0
F1 Score = 1.0

Code :

I here attached my Github Repository for this assignment 04 as EN_Ass4

https://github.com/Garthigan/EN_Ass4