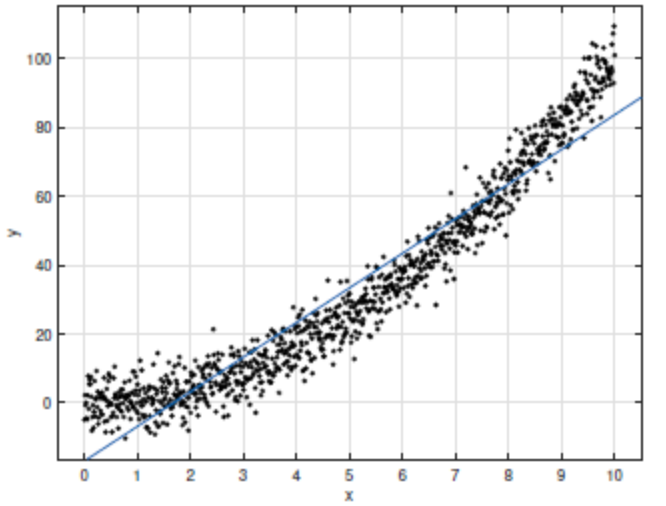
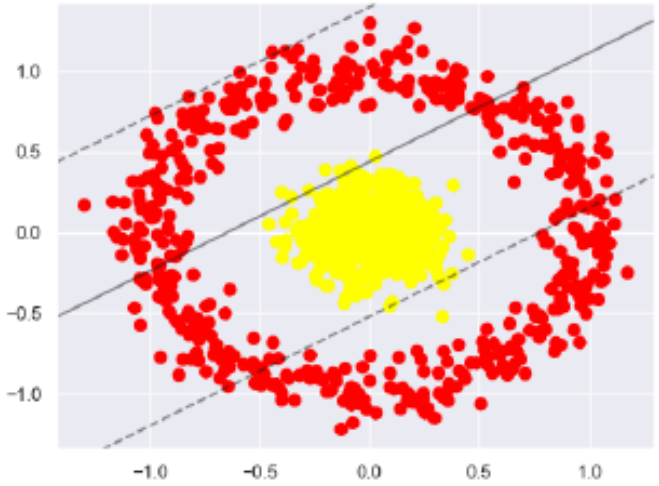
Lecture 7 (kernel methods)

Limitations of Linear Classification

- In the previous lecture, we have learnt of how to classify different datapoints by using a linear classifier (perceptron, linear svm). we have learnt of how to learn different relationships between variables using linear regression, or how to classify different datapoints by using a linear classifier (perceptron, linear svm).
- **What happens if the relationship between the variables is non-linear as shown in Fig. 1(a)?**
- **What happens if the different datapoins can not be separated by linear classifier as shown in Fig. 1(b)?**



(a) Linear Regression on non-linear data

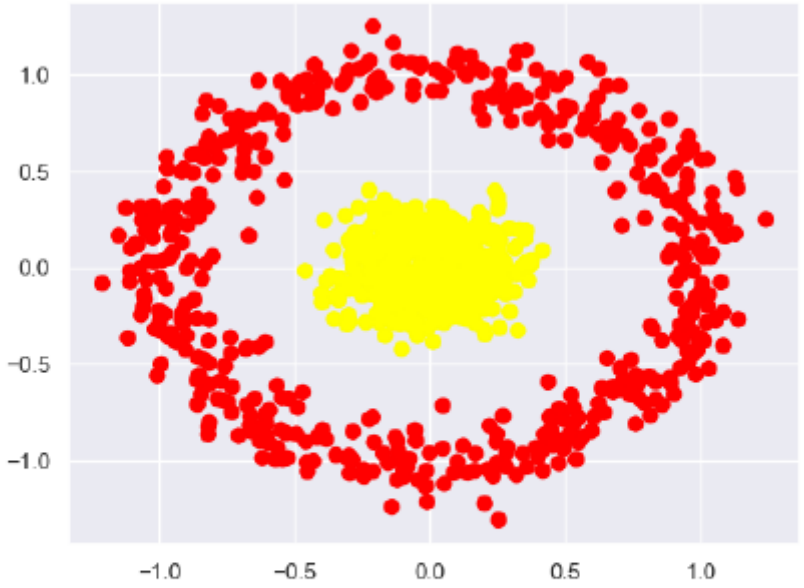


(b) Linear svm on non-linearly separable data

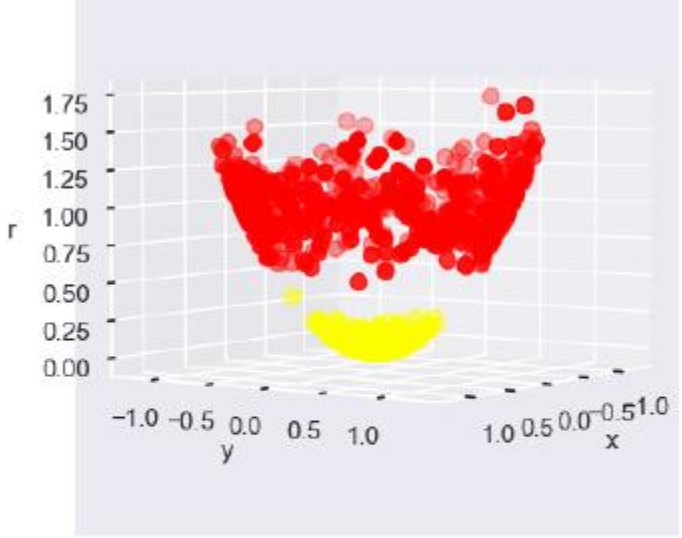
Non-Linear Classification

- In this lecture, we will explore the idea of non-linear classification. The idea that we will use is that we will transform the data $x \in \mathbb{R}^{d_1}$ to $\phi(x) : \mathbb{R}^{d_1} \rightarrow \mathbb{R}^{d_2}$, where $d_2 > d_1$, and the transformed data will be linearly separable.

higher dimension



(a) Non-linearly separable data



(b) Linearly Separable Transformed Data

Non-Linear Classification

- In the previous example we have

$$\boldsymbol{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \in \mathbb{R}^2$$

The data is not linearly separable and so to overcome this issue we will do the following transformation:

$$\phi(\boldsymbol{x}) = \begin{bmatrix} x_1 \\ x_2 \\ x_1^2 + x_2^2 \end{bmatrix} \in \mathbb{R}^3$$

After the new transformation, the data is linearly separable. The classifier will have the following form:

$$h(\phi(\boldsymbol{x}); \theta, \theta_0) = \text{sgn}(\theta^T \phi(\boldsymbol{x}) + \theta_0) = \text{sgn}(\theta_1 x_1 + \theta_2 x_2 + \theta_3 (x_1^2 + x_2^2) + \theta_0)$$

weights, biases, transformed x, classification

Non-Linear Classification

- The classifier that we will get will be a circle in 2D as shown in figure below

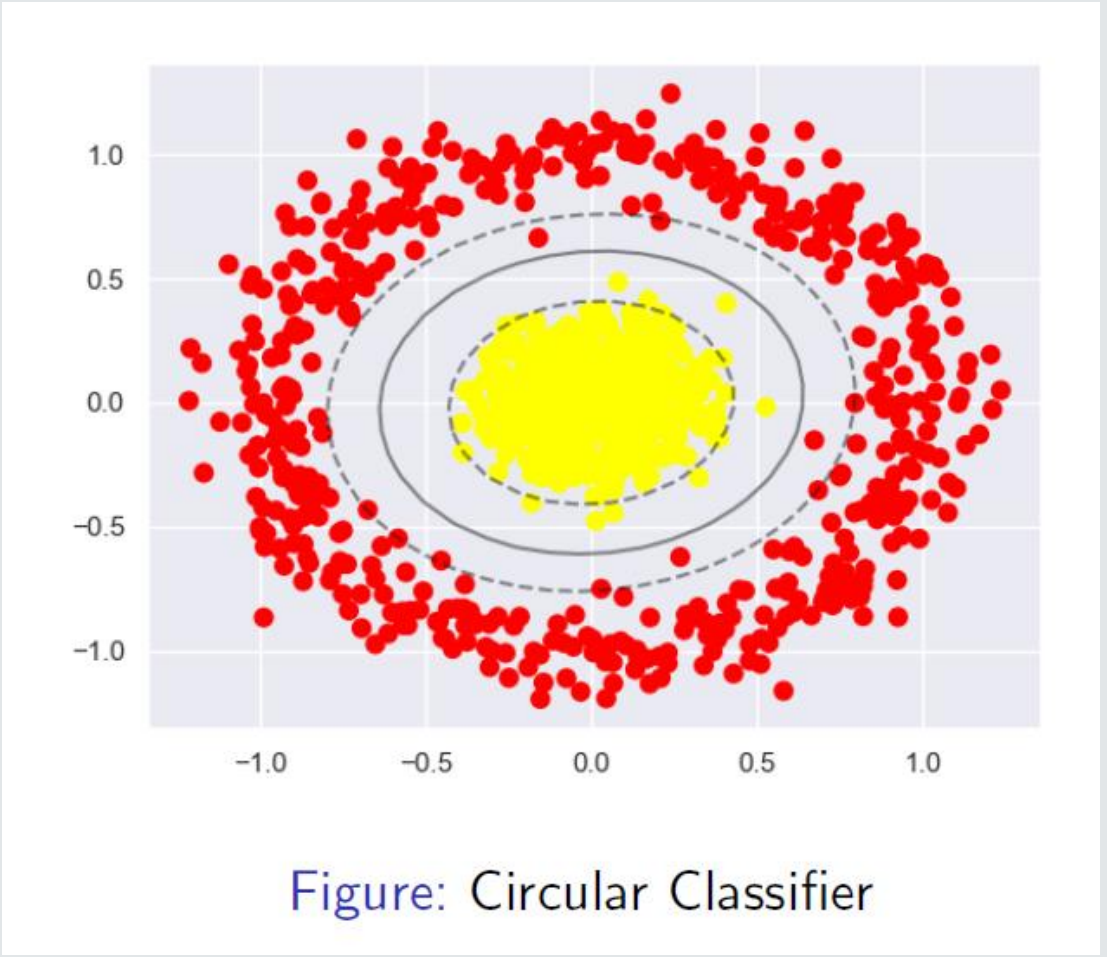


Figure: Circular Classifier

Kernel Functions

- As we have seen that these functions will map the data to higher dimensions and this results in larger vectors and so higher complexity. Therefore, we will discuss two specific functions we can compute efficiently.
- **Polynomial function**: Assume we have $\boldsymbol{x} \in \mathbb{R}^n$ and the polynomial of degree d:

$$\phi(\boldsymbol{x}) = \left[\frac{\sqrt{d!}}{\sqrt{j_1! j_2! \cdots j_{n+1}!}} x_1^{j_1} \cdots x_n^{j_n} 1^{j_{n+1}} \right]_{j_1 + j_2 + \cdots + j_{n+1} = d}$$

For example if we have $n = 2, d = 2$

$$\phi(\boldsymbol{x}) = \begin{bmatrix} 1\,(j_1 = 0, j_2 = 0, j_3 = 2) \\ \sqrt{2}x_1(j_1 = 1, j_2 = 0, j_3 = 1) \\ \sqrt{2}x_2(j_1 = 0, j_2 = 1, j_3 = 1) \\ \sqrt{2}x_1x_2(j_1 = 1, j_2 = 1, j_3 = 0) \\ x_1^2(j_1 = 2, j_2 = 0, j_3 = 0) \\ x_2^2(j_1 = 0, j_2 = 2, j_3 = 0) \end{bmatrix}$$

Kernel Functions

- The size of the vector $\phi(\boldsymbol{x})$ is

$$\binom{n+d}{d} = \frac{(n+d)!}{d!\,n!}$$

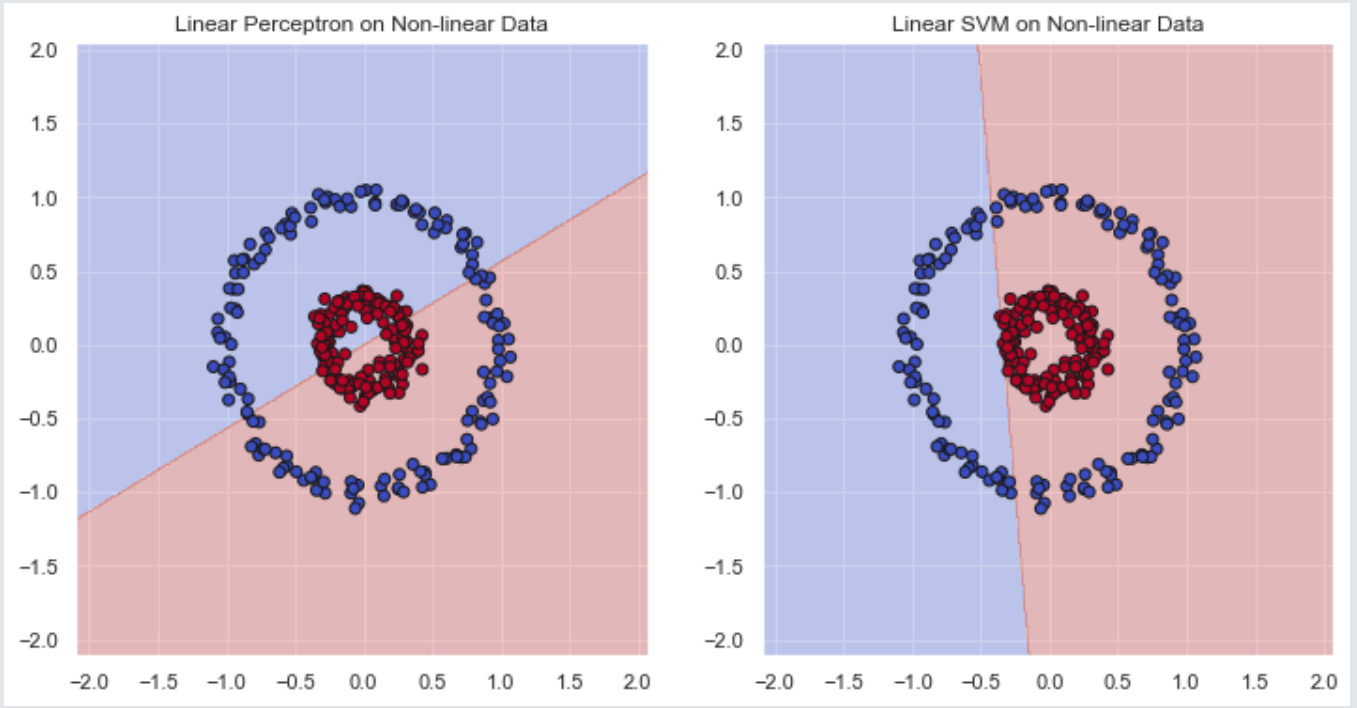
If we have a large n or d or both then $\phi(\boldsymbol{x})$ can get very large and become very expensive to deal with

$$n = 20, d = 2 \Rightarrow \binom{22}{2} = 231$$

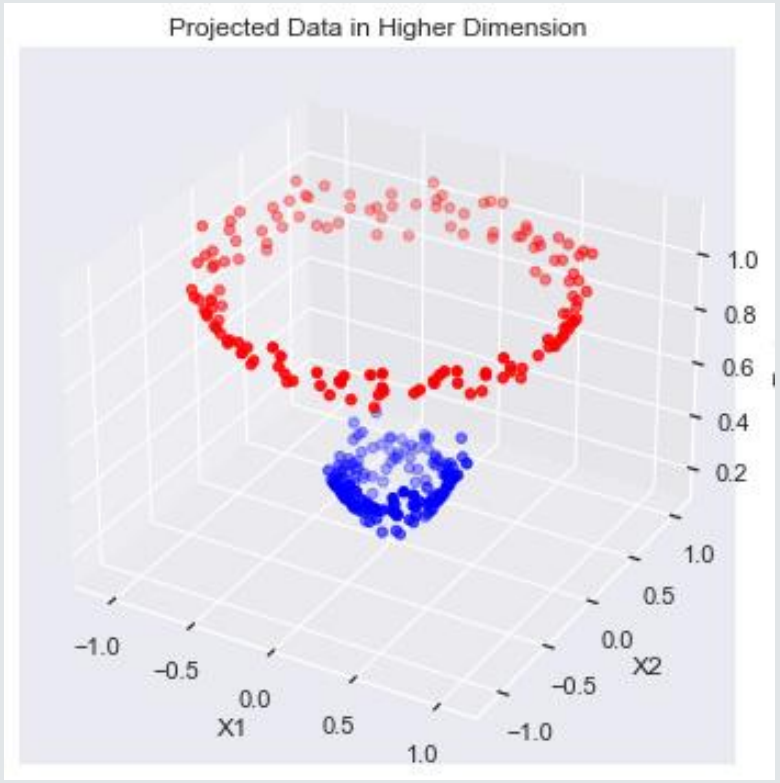
$$n = 100, d = 2 \Rightarrow \binom{102}{2} = 5151$$

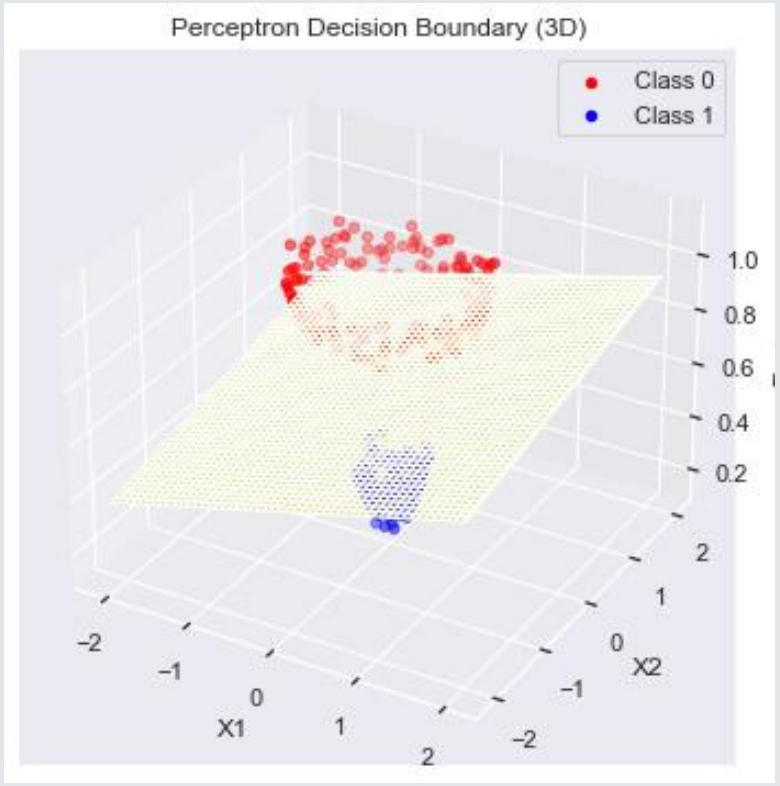
$$n = 100, d = 3 \Rightarrow \binom{103}{3} = 176851$$

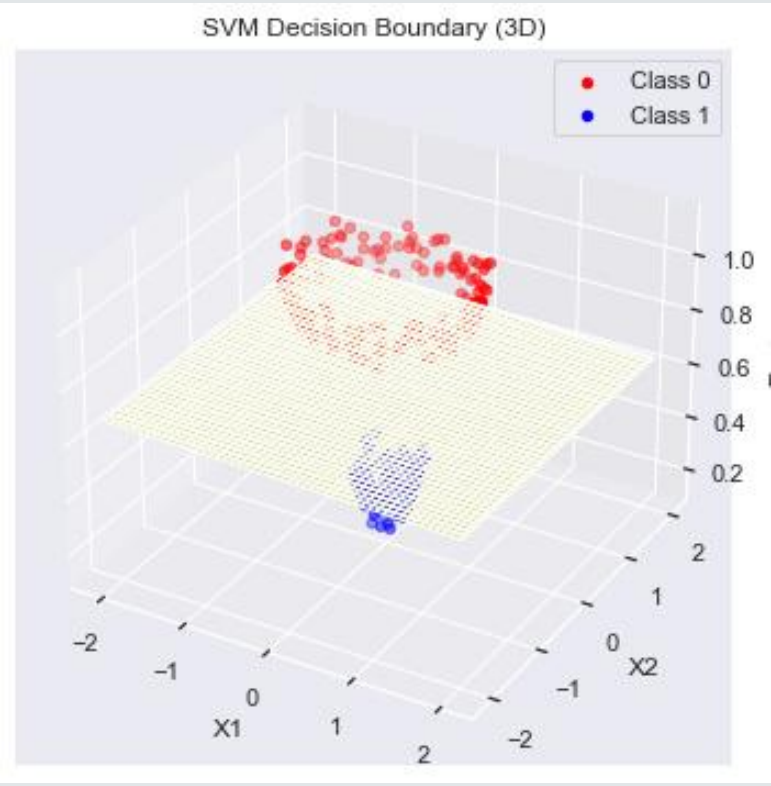
Non-Linear Classification (Experiment 3)



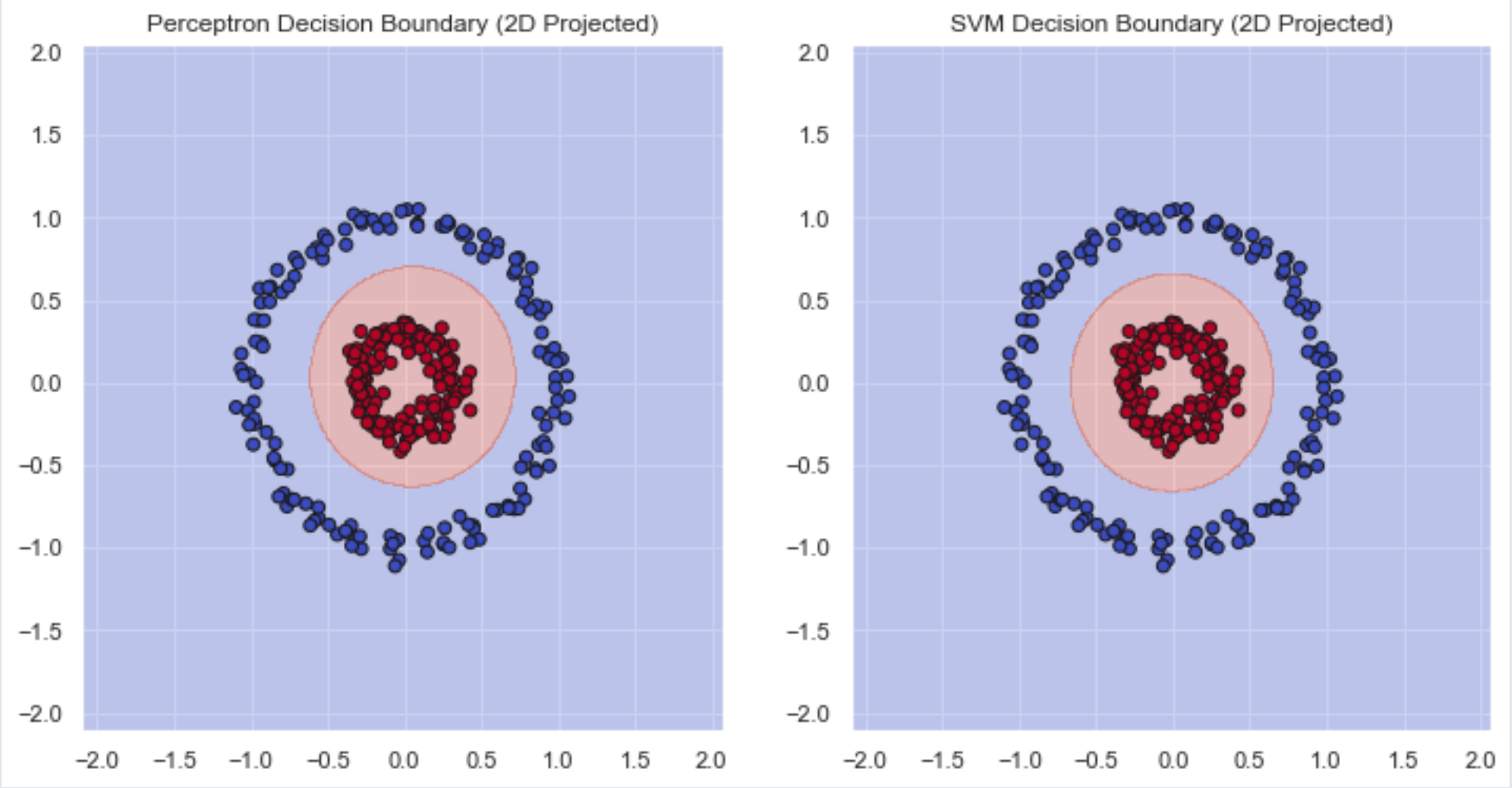
Non-Linear Classification (Experiment 3)







Non-Linear Classification (Experiment 3)



Kernel Functions

- **Additivity**: The sum of two valid kernels is also a valid kernel.
- **Scalar Multiplication**: The product of a valid kernel and a positive scalar is also a valid kernel.
- **Product of Kernels**: The product of two valid kernels is also a valid kernel.
- **Exponentiation**: Raising a valid kernel to a positive power yields another valid kernel.

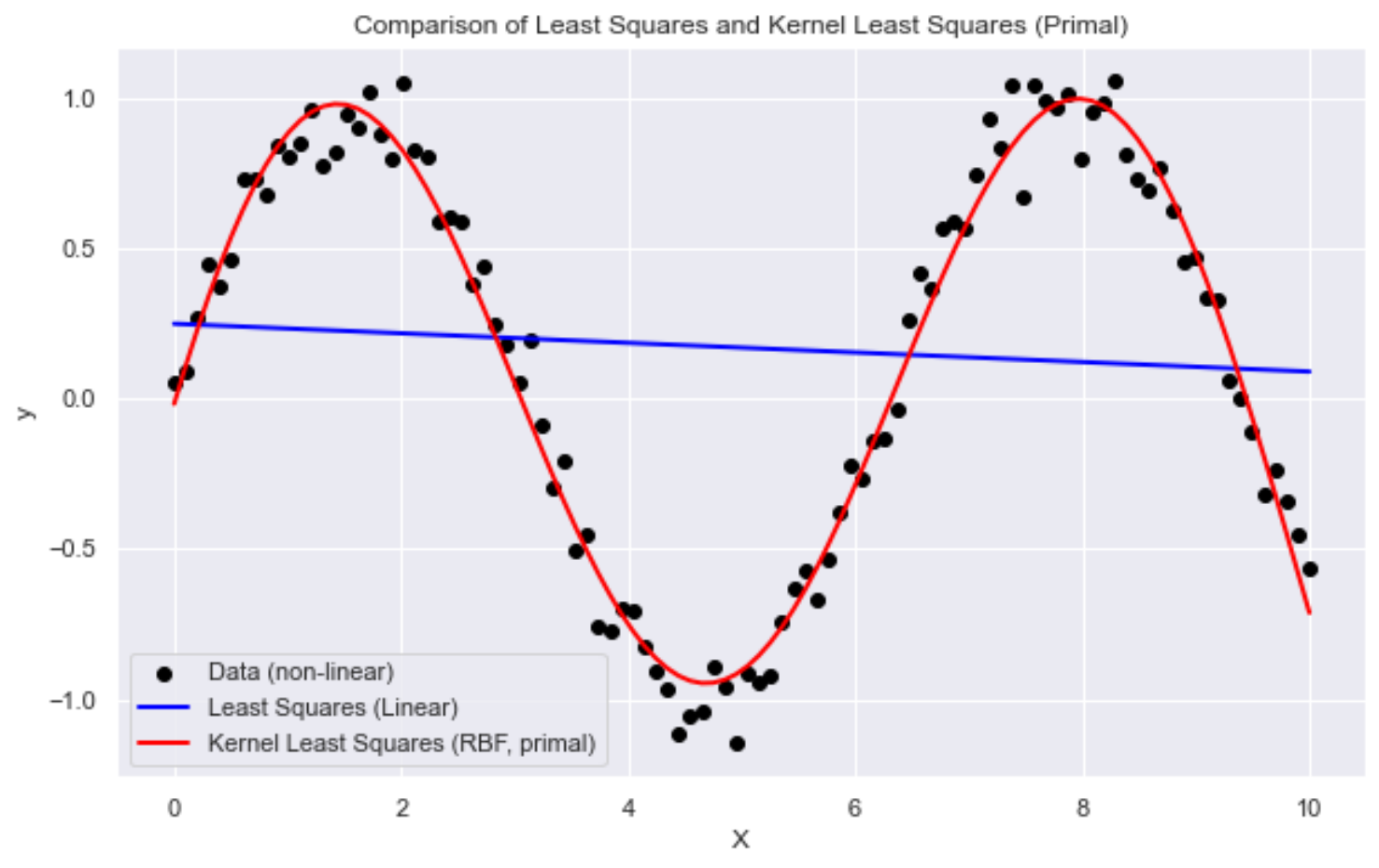
Kernel Least Square

- In linear least squares, the model is limited to linear relationships between input features and outputs. To handle non-linear relationships, we can map the input data into a higher-dimensional space using a feature map $\phi(x)$, which transforms $x \in \mathbb{R}^d$ to a higher-dimensional space.

$$\min_{\beta} \left|\left|y - \phi(X)\beta\right|\right|_2^2$$

$$\boldsymbol{\beta = \left(\phi(X)^T \phi(X)\right)^{-1} \phi(X)^T y}$$

Kernel Least Square (Experiment 4)



Limitations of Kernel Least Square

- **Complex Patterns:** When data is too complex, kernel methods might not capture the intricacies
- **Overfitting:** Flexible kernels can easily overfit, especially in noisy datasets.
- **Curse of Dimensionality:** Kernel methods may become ineffective in very high-dimensional spaces.
- **Kernel and Parameter Tuning:** Choosing the right kernel and tuning parameters is often non-trivial.