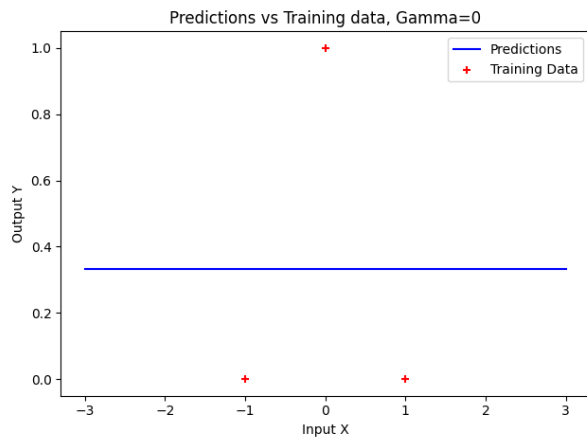# Week 6 Assignment

Efeosa Eguavoen - 17324649
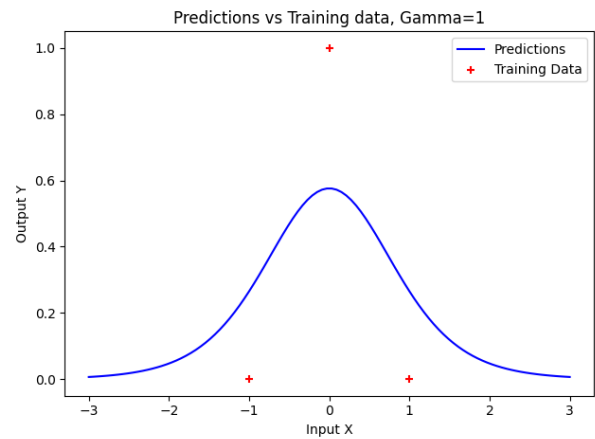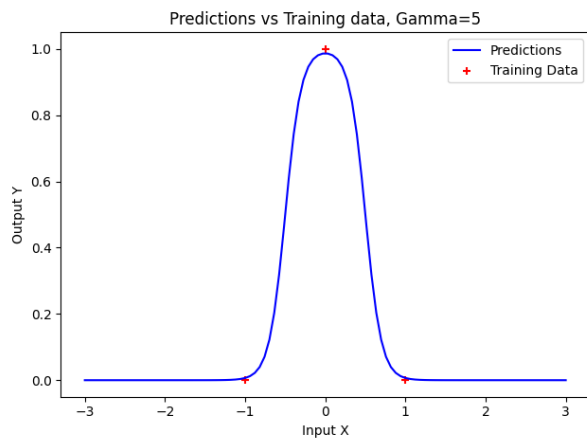
November 17, 2020
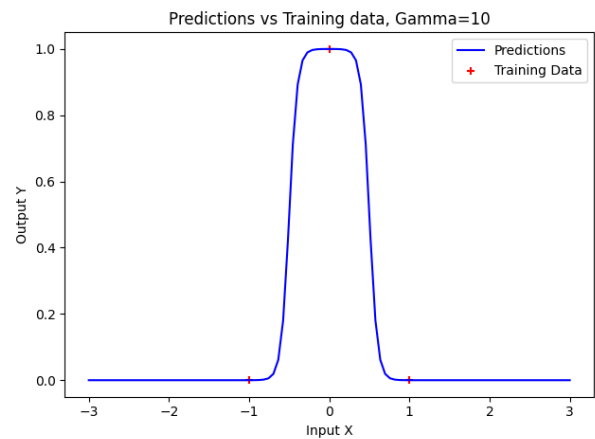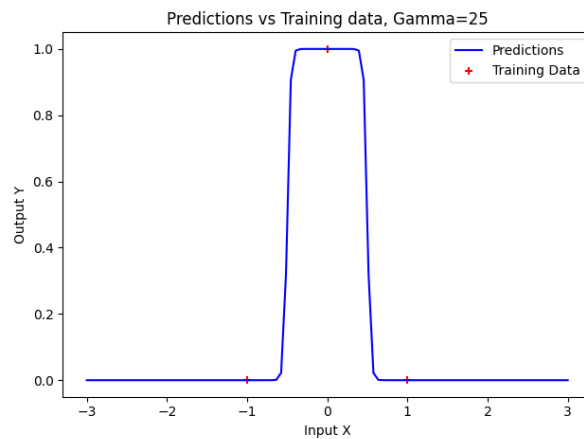
# 1  (i)

## 1.1  A



(a) Gamma = 0

(b) Gamma = 1

(c) Gamma = 5

(d) Gamma = 10

(e) Gamma = 25

From the above figures we can see that as we increase gamma, our predictions fit closer to the training data, with low values of data underfitting the data such as when gamma = 1 and higher values of gamma confirming very closely to the data such as when gamma = 25.

## 1.2  B

From the graphs in A we can see our predictions changing as we increase the value of Gamma. This is due to fact that as we increase gamma, we decrease the rate at which the out put for our kernel grows. KNN uses the distances between a point and all other points in the data and selects the K closest examples, then sets the averages of the labels. Normally all datapoints have uniform weights but by using a gaussian kernel, the weights of datapoints changes as they get further away from the origin datapoint, with gamma effecting how quickly this change happens. From our data we can see that small values of Gamma make the kernel decrease much slower with distance and underfit the data, hence the slopes we see when Gamma = 1, due to less weight being on points from the origin point. In contrast, high values of Gamma make the kernel decrease very quickly, putting more weight on points even if they're very close to the origin, causing the preditions to snap towards the nearest datapoint such as when Gamma = 25. This also explains why when x is between -.5 and .5 we see y =1 as it's snapping towards the nearest data point.

## 1.3  C

From the graphs below, we can see that as we change C and the Gamma, the shape of our graph alters quite considerably. Higher C values gives graphs with higher peaks and with higher gamma values influencing this also.

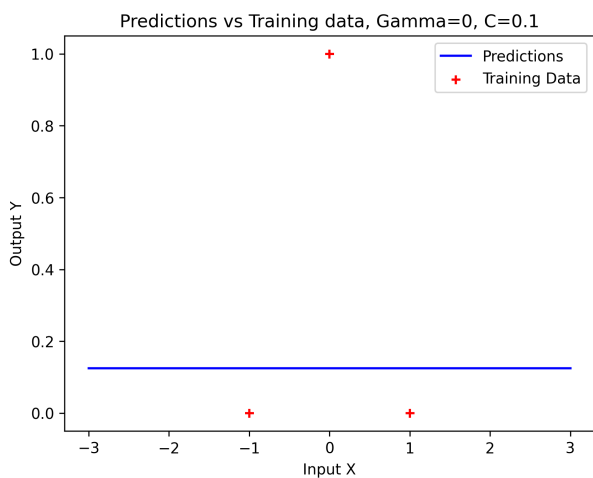The below values represent each page of graphs hence the blocks of 5.

$\theta_0 = -0.025, \theta_1 = 0.175, \theta_2 = -0.025$
$\theta_0 = -0.0103, \theta_1 = 0.1679, \theta_2 = -0.0103$
$\theta_0 = -0.0002, \theta_1 = 0.1667, \theta_2 = -0.0002$
$\theta_0 = -0.0, \theta_1 = 0.1667, \theta_2 = -0.0$
$\theta_0 = -0.0, \theta_1 = 0.1667, \theta_2 = -0.0$

$\theta_0 = -0.5714, \theta_1 = 1.4286, \theta_2 = -0.5714$
$\theta_0 = -0.1833, \theta_1 = 0.7566, \theta_2 = -0.1833$
$\theta_0 = -0.003, \theta_1 = 0.6667, \theta_2 = -0.003$
$\theta_0 = -0.0, \theta_1 = 0.6667, \theta_2 = -0.0$
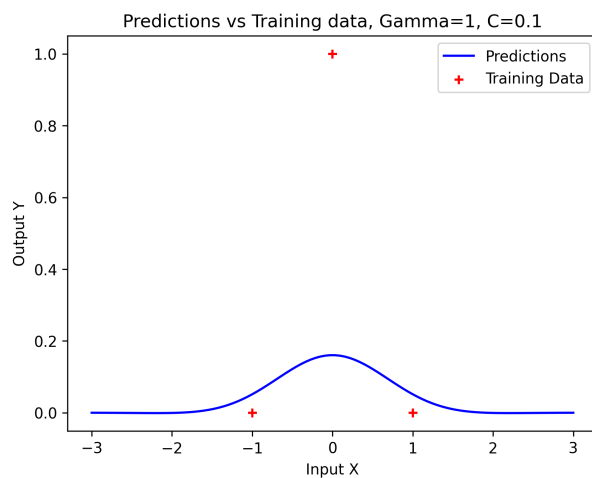$\theta_0 = -0.0, \theta_1 = 0.6667, \theta_2 = -0.0$

$\theta_0 = -6.5574, \theta_1 = 13.4426, \theta_2 = -6.5574$
$\theta_0 = -0.4323, \theta_1 = 1.2553, \theta_2 = -0.4323$
$\theta_0 = -0.0061, \theta_1 = 0.9525, \theta_2 = -0.0061$
$\theta_0 = -0.0, \theta_1 = 0.9524, \theta_2 = -0.0$
$\theta_0 = -0.0, \theta_1 = 0.9524, \theta_2 = -0.0$

$\theta_0 = -66.5557, \theta_1 = 133.4443, \theta_2 = -66.5557$
$\theta_0 = -0.4855, \theta_1 = 1.3504, \theta_2 = -0.4855$
$\theta_0 = -0.0067, \theta_1 = 0.9951, \theta_2 = -0.0067$
$\theta_0 = -0.0, \theta_1 = 0.995, \theta_2 = -0.0$
$\theta_0 = -0.0, \theta_1 = 0.995, \theta_2 = -0.0$
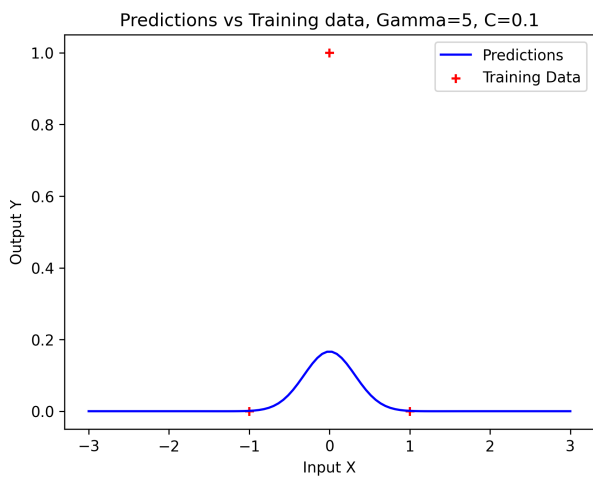
$\theta_0 = -666.5556, \theta_1 = 1333.4444, \theta_2 = -666.5556$
$\theta_0 = -0.4914, \theta_1 = 1.3609, \theta_2 = -0.4914$
$\theta_0 = -0.0067, \theta_1 = 0.9996, \theta_2 = -0.0067$
$\theta_0 = -0.0, \theta_1 = 0.9995, \theta_2 = -0.0$
$\theta_0 = -0.0, \theta_1 = 0.9995, \theta_2 = -0.0$
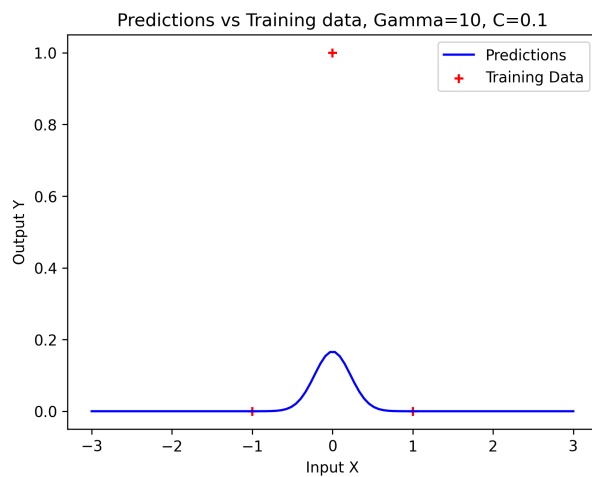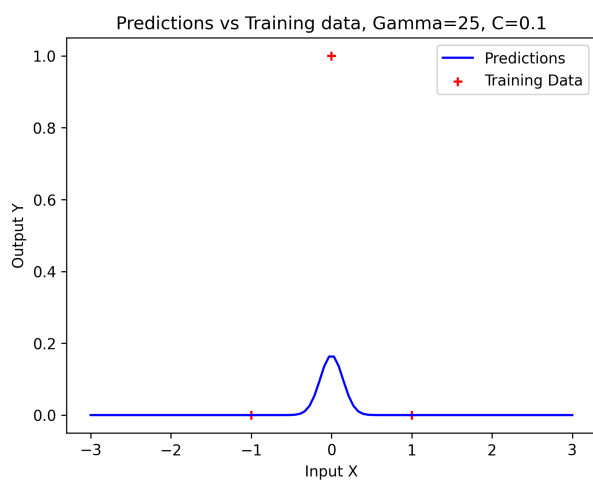
(a) Gamma = 0, C = 0.1

(b) Gamma = 1, C = 0.1

(c) Gamma = 5, C= 0.1
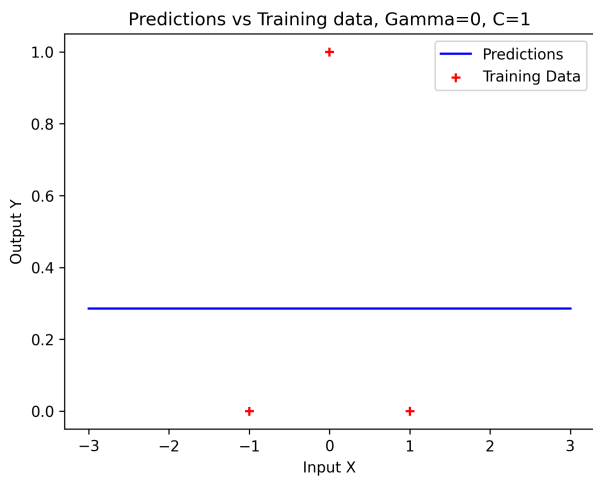
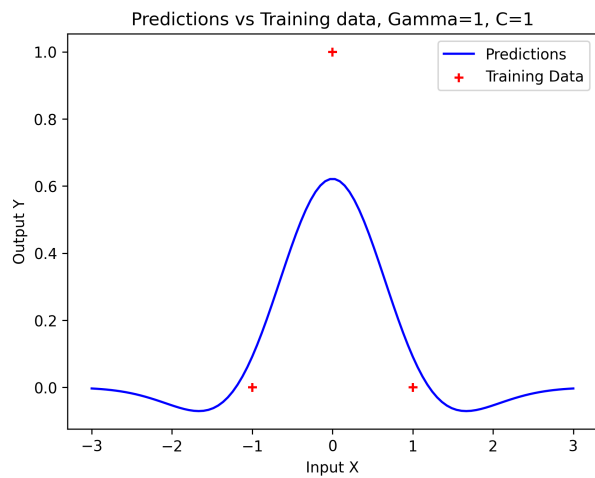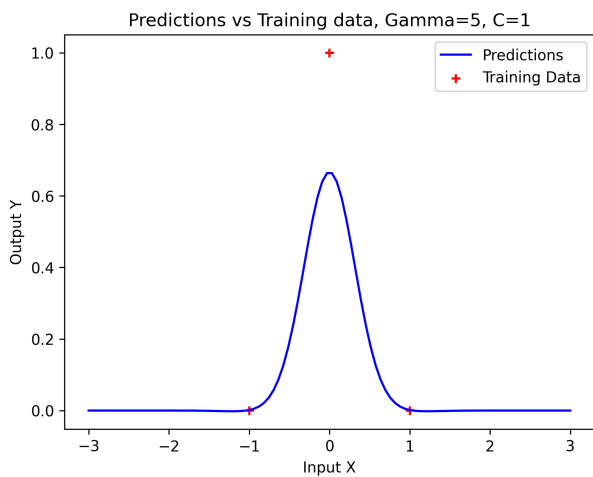(d) Gamma = 10, C= 0.1

(e) Gamma = 25, C= 0.1
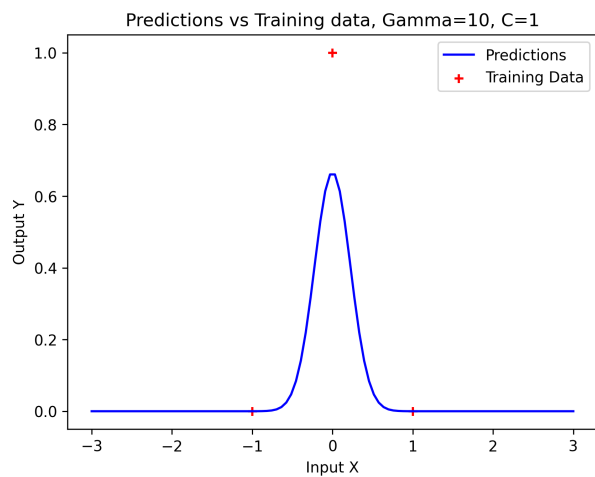
(a) Gamma = 0, C = 1
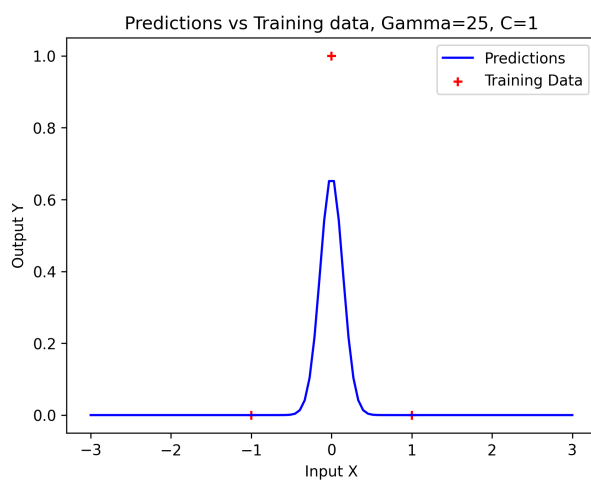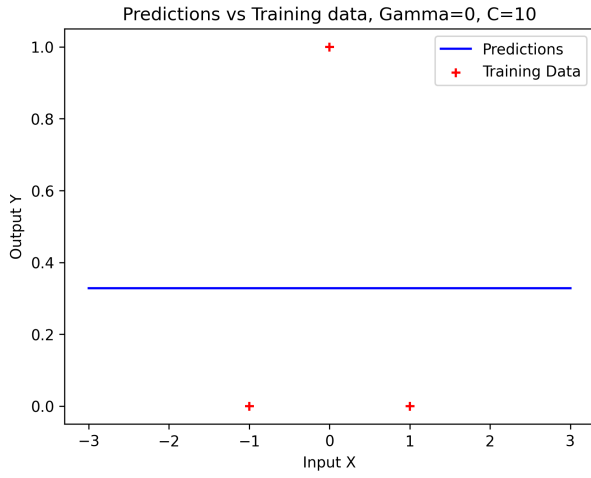
(b) Gamma = 1, C = 1

(c) Gamma = 5, C = 1

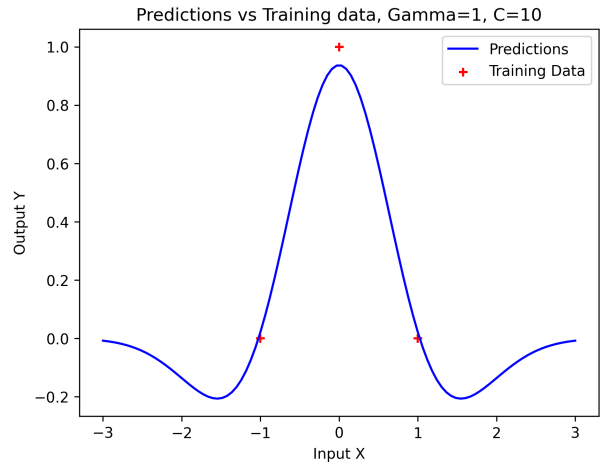(d) Gamma = 10, C = 1

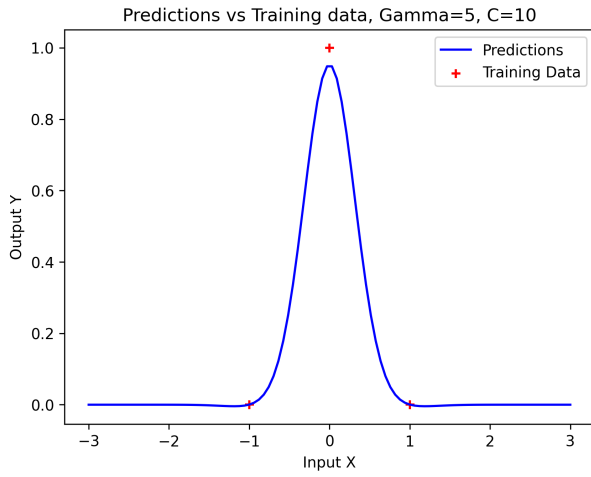(e) Gamma = 25, C = 1

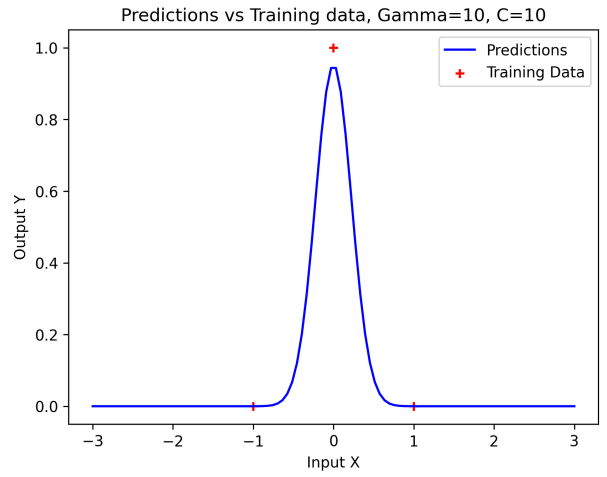(a) Gamma = 0, C = 10

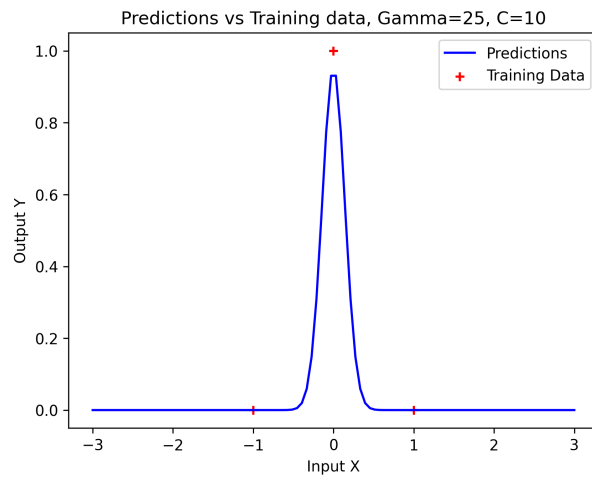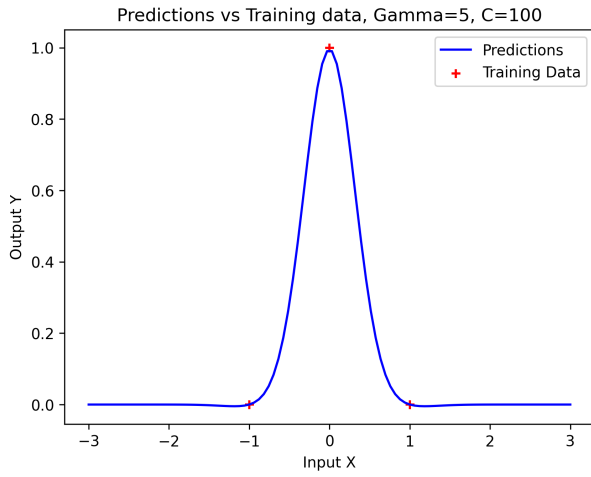(b) Gamma = 1, C = 10

(c) Gamma = 5, C= 10

(d) Gamma = 10, C= 10
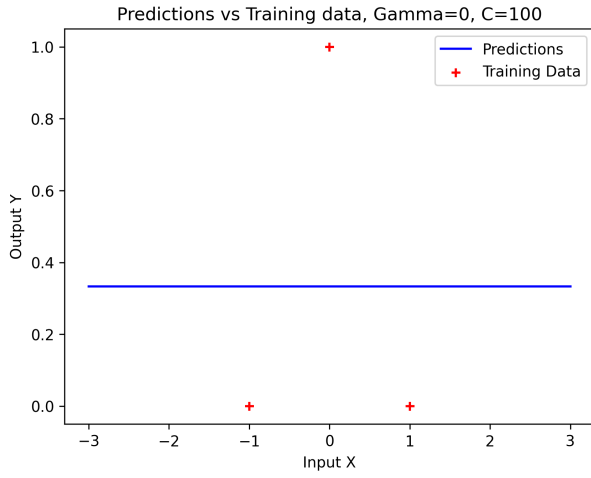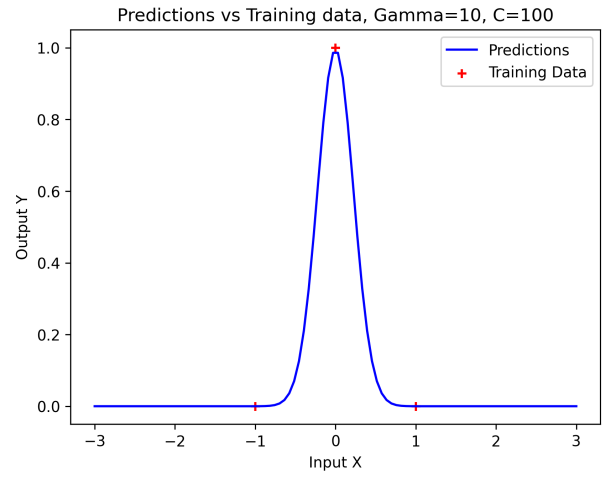
(e) Gamma = 25, C= 10
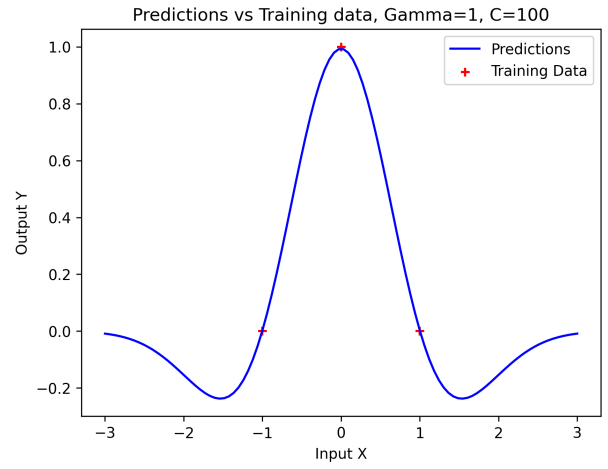
(a) Gamma = 0, C = 100

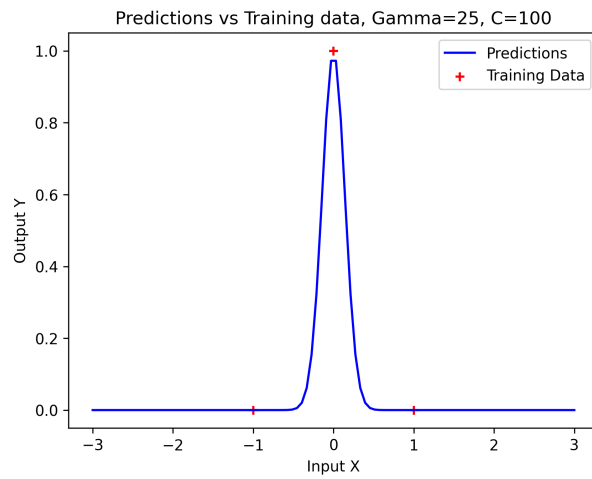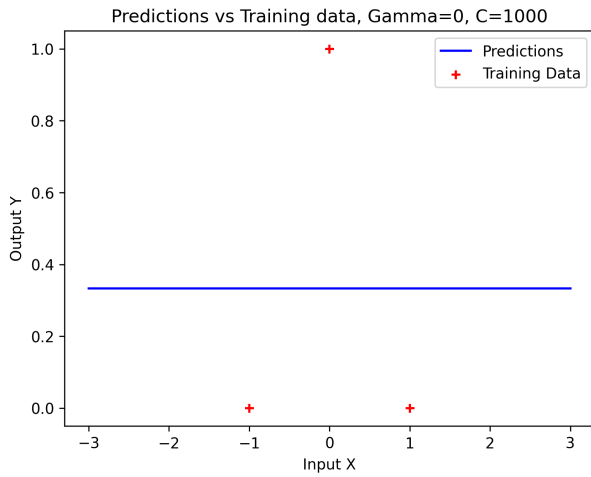(b) Gamma = 1, C = 100

(c) Gamma = 5, C = 100
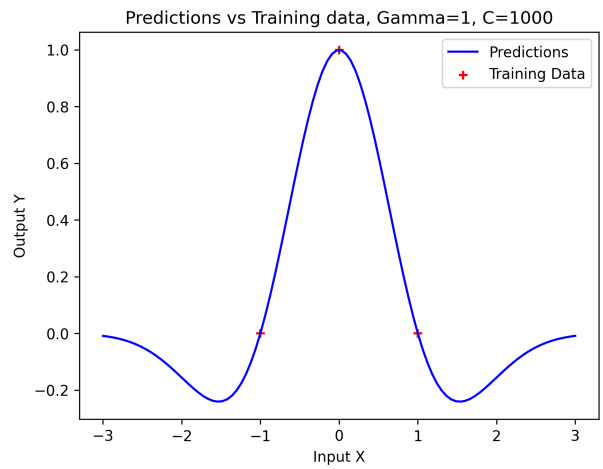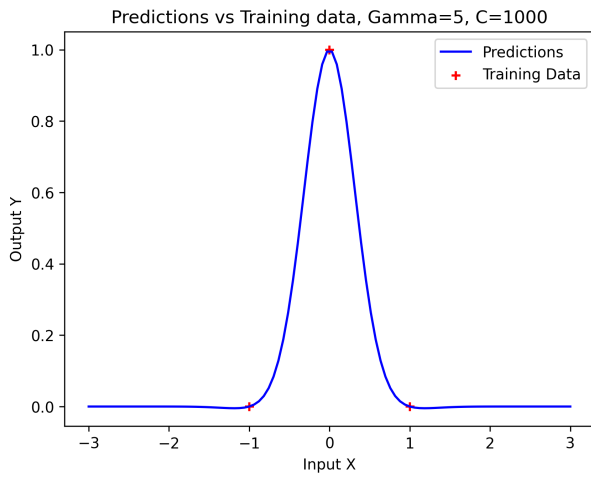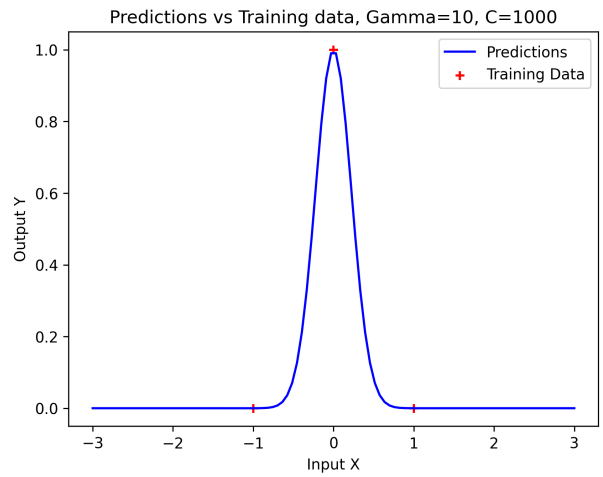
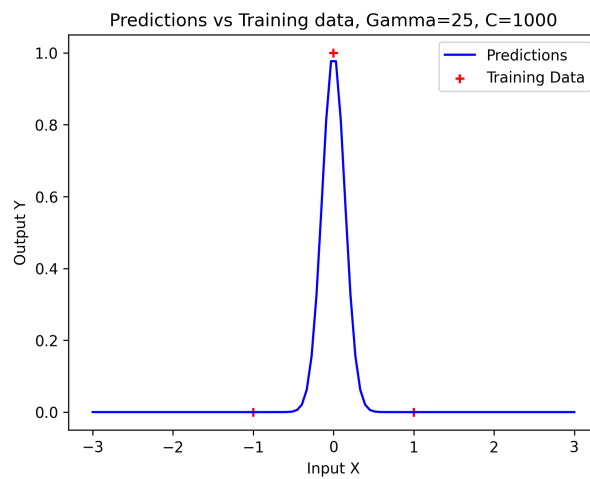(d) Gamma = 10, C= 100

(e) Gamma = 25, C= 100

(a) Gamma = 0, C = 1000

(b) Gamma = 1, C = 1000

(c) Gamma = 5, C= 1000

(d) Gamma = 10, C= 1000

(e) Gamma = 25, C= 1000

## 1.4  D

Kernalised Ridge Regression works by applying a L2 penalty to normal linear regression to shrink parameters down to near 0 that aren't as important to the model in an effort to prevent overfitting. This used along side a kernel can introduce non-linearity to the regressor by introducing new features that are non-linear depending on the type of kernel used. In this case we use a Gaussian kernel to introduce new features hence enabling us to fit a non-linear curve to our data. When we vary our values of C and Gamma, we alter how much of a penalty does the L2 penalty give and also alter how much each of our new features introduced by our kernel effects the slope of our curve. From our data we can see that lower values of Gamma give us much more curved predictions while higher values of gamma snap to the data points. Lower values of C allow for much more of our data to be underfitted while higher values of C do not. When we take these 2 together, Gamma gives us the slope of our line due to introducing non-linear features while C reigns in the slope by giving a penalty to parameters that are not as important in dictating the slope of our line. We can see this in our data, as our value of Gamma is increasing, C gives a more strict penalty hence why more of out parameters are approaching 0(I've rounded the data). In terms of behaviour between KNN and Kernalised Ridge Regression, Kernalised Ridge Regression allows for more flexible models overall as we can tune both C and Gamma, whiles in KNN we can only tune Gamma. This leads to better predictions as we can get more general and flexible models. KNN seems to much more readily snap to data and give a range of data the same Y output whilst for the same value of Gamma in KRR, it's more general.