Supervised Learning Cw1

22063614 22084226

November 16, 2022

1 Part1

1.1 Linear Regression

1.1.1 Question 1

(a) Plot graphs for four dimensions k = 1,2,3,4 and plot four points on them, see Figure 1

(b) Equations for curves fitted for k = 1,2,3 are:

$$when\, k = 1, y = 2.5$$

$$when\, k = 2, y = 1.5 + 0.4x$$

$$when\, k = 3, y = 9 - 7.1x + 1.5x^2$$

(c) For each curve k = 1,2,3,4 the mean square errors are:

$$when\, k = 1, MSE = 3.25$$

$$when\, k = 2, MSE = 3.05$$

$$when\, k = 3, MSE = 0.8$$

$$when\, k = 4, MSE = 3.4942940895563015e - 23$$

1.1.2 Question 2

(a) i Plot function $sin^2(2\pi x)$ in the range $0 \leq x \leq 1$ and plot generated points, see Figure 2

ii Plot 5 curves for dimension k = 2,5,10,14,18, with data points see Figure 3

(b) Plot the ln of the training error versus the polynomial dimension k = 1 to 18, see Figure 4

(c) Plot the ln of the test error versus the polynomial dimension k = 1 to 18, see Figure 5

(d) Plot the average results of a 100 runs of items (b) and (c), see Figure 6

1.1.3 Question 3

(b) Plot the ln of the training error versus the polynomial dimension k = 1 to 18, see Figure 7

(c) Plot the ln of the test error versus the polynomial dimension k = 1 to 18, see Figure 8

(d) Plot the average results of a 100 runs of items (b) and (c), see Figure 9

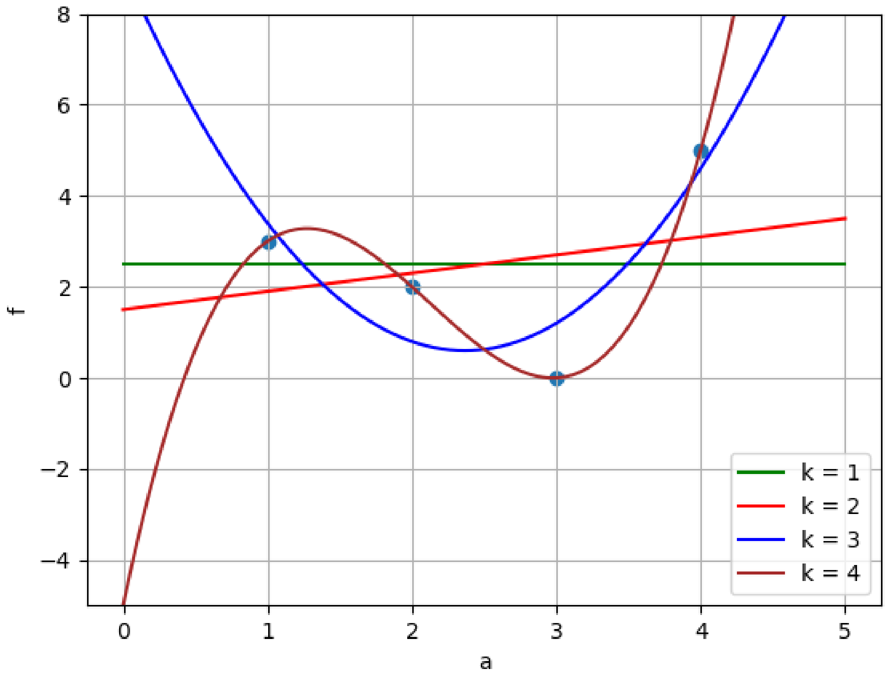

Figure 1: plot four curves of four dimensions

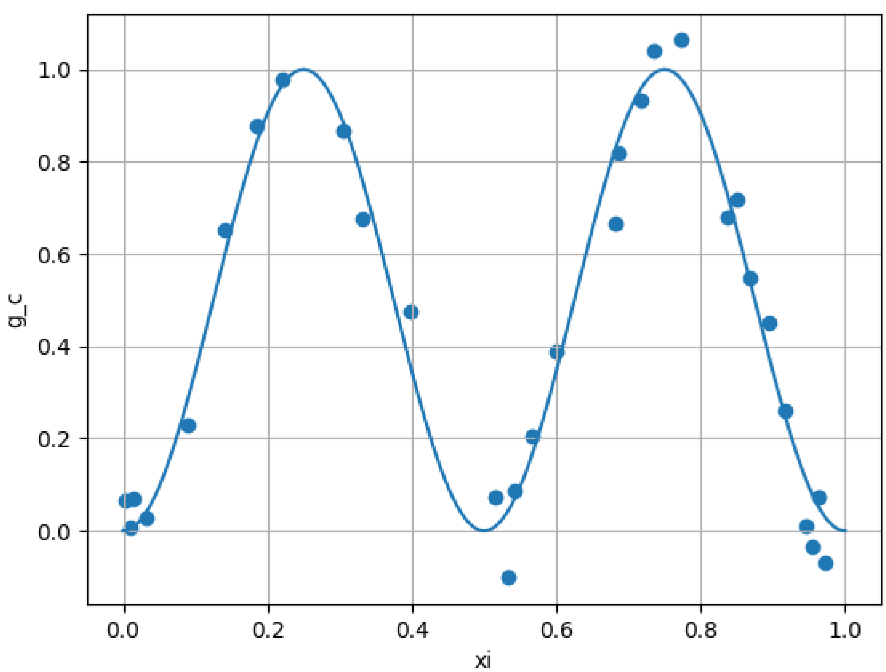

Figure 2: Plot curve in range $0 \leq x \leq 1$ with data points

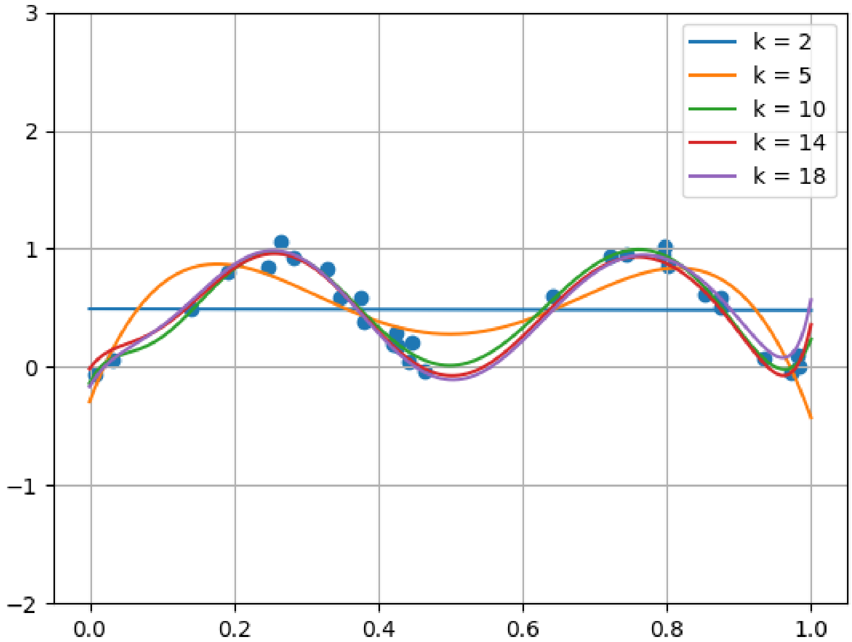

Figure 3: Plot curves for k = 2,5,10,14,18 with data points

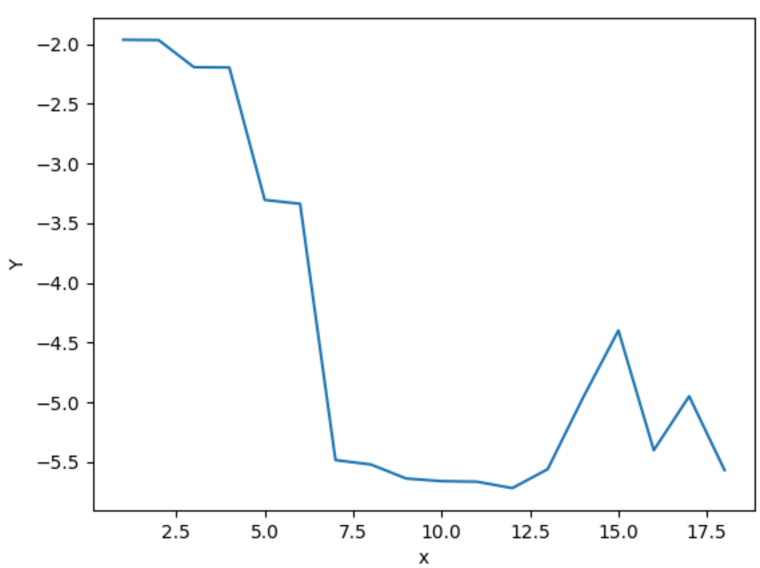

Figure 4: Plot curves for training error versus k = 1 to 18

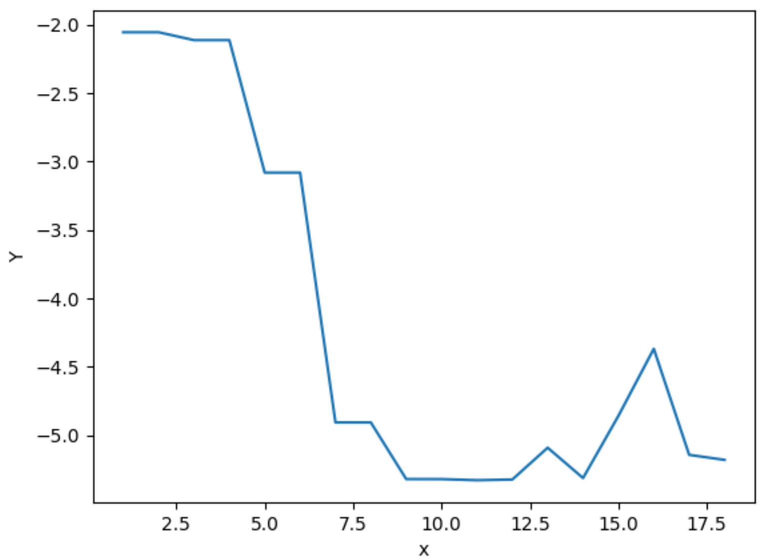

Figure 5: Plot curves for test error versus k = 1 to 18

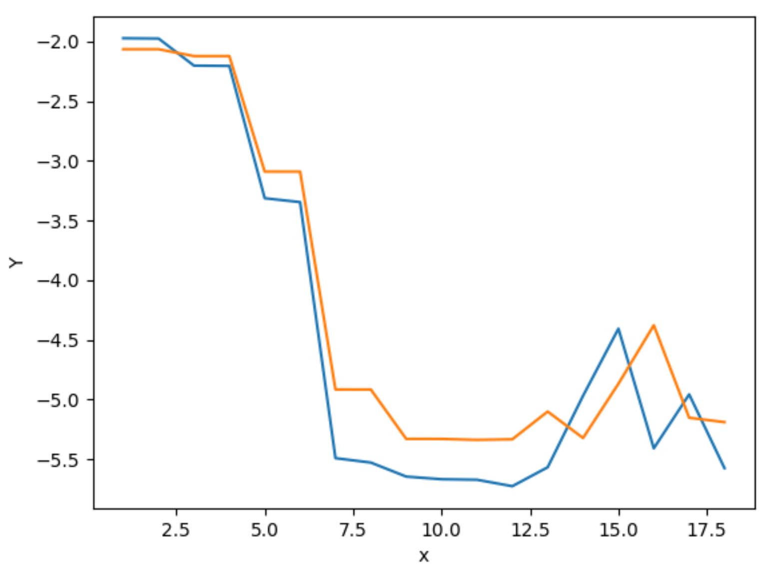

Figure 6: Training error(blue) and test error(yellow) versus k = 1 to 18

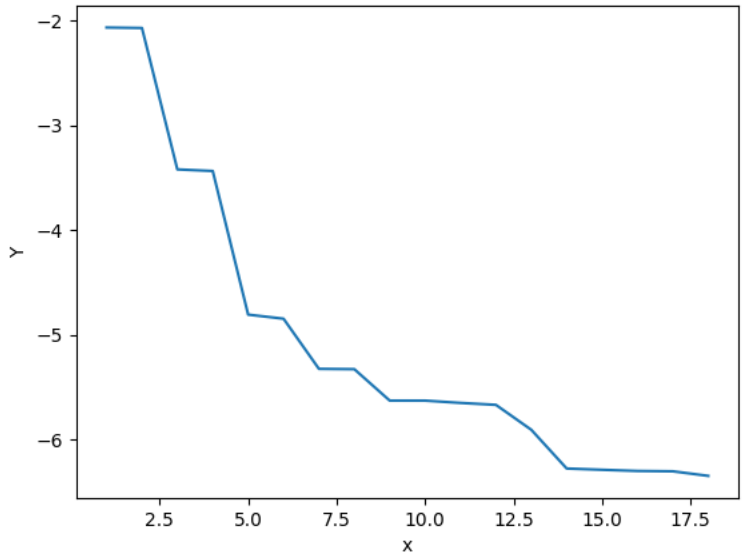

Figure 7: Plot curves for training error versus k = 1 to 18

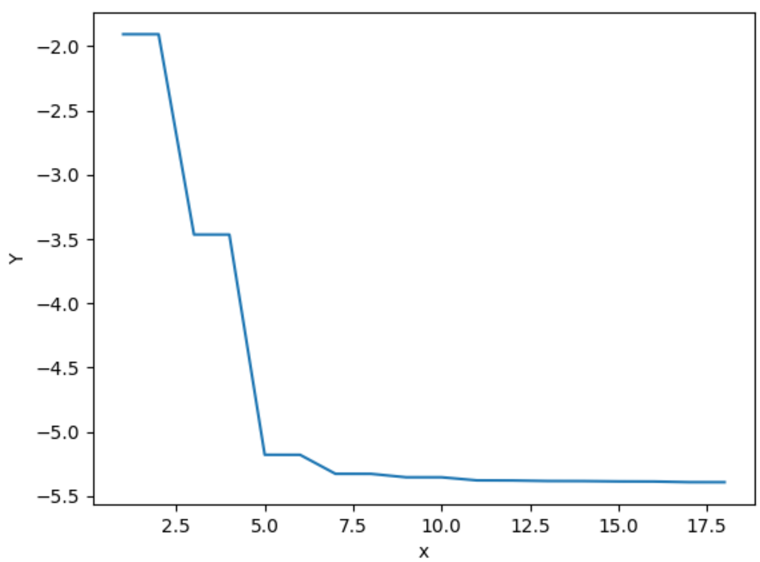

Figure 8: Plot curves for test error versus k = 1 to 18

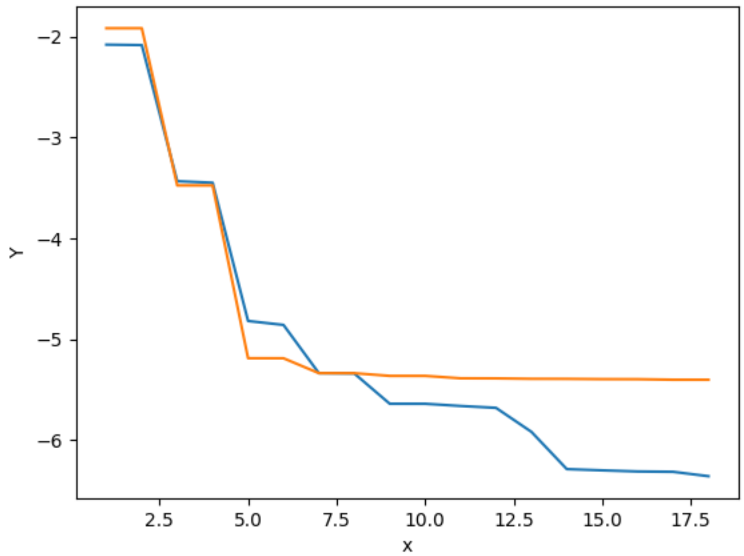

Figure 9: Training error(blue) and test error(yellow) versus k = 1 to 18

1.1.4 Question 4

(a) The MSE for training set is 84.14630111119814
The MSE for test set is 85.34445599406038

(b) The constant function uses the same x value for all y, which means it is used to calculate the mean value of y.

(c)
- MSE for training set of Linear Regression with attribute 1 is 71.28870008291257 MSE for test set of Linear Regression with attribute 1 is 73.16559303613107
- MSE for training set of Linear Regression with attribute 2 is 72.41162303578434 MSE for test set of Linear Regression with attribute 2 is 76.12855911497527
- MSE for training set of Linear Regression with attribute 3 is 64.32155905484075 MSE for test set of Linear Regression with attribute 3 is 65.91357141111934
- MSE for training set of Linear Regression with attribute 4 is 81.12945739614695 MSE for test set of Linear Regression with attribute 4 is 84.26280597141928
- MSE for training set of Linear Regression with attribute 5 is 68.2121982443285 MSE for test set of Linear Regression with attribute 5 is 71.05182632081701
- MSE for training set of Linear Regression with attribute 6 is 43.720393923772995 MSE for test set of Linear Regression with attribute 6 is 43.84588696805582
- MSE for training set of Linear Regression with attribute 7 is 71.4869410503565 MSE for test set of Linear Regression with attribute 7 is 74.8189445367801
- MSE for training set of Linear Regression with attribute 8 is 78.23127521815532 MSE for test set of Linear Regression with attribute 8 is 81.59529458654262
- MSE for training set of Linear Regression with attribute 9 is 71.52229703414014 MSE for test set of Linear Regression with attribute 9 is 73.90468057477446
- MSE for training set of Linear Regression with attribute 10 is 65.47845959851624 MSE for test set of Linear Regression with attribute 10 is 67.19909810392437
- MSE for training set of Linear Regression with attribute 11 is 62.36887025011994 MSE for test set of Linear Regression with attribute 11 is 63.7265256860768

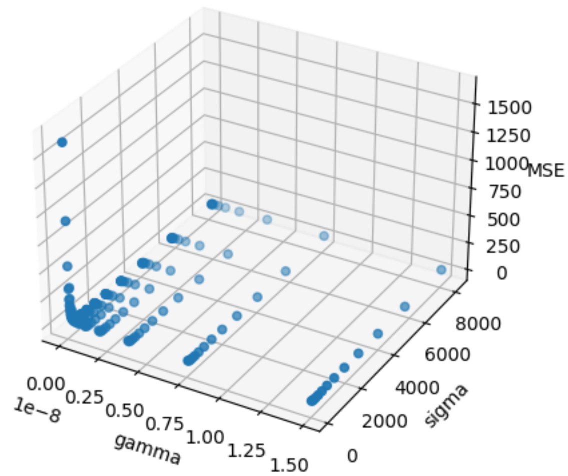

Figure 10: mean value over folds of validation error versus σ and γ

- MSE for training set of Linear Regression with attribute 12 is 37.905237494552175 MSE for test set of Linear Regression with attribute 12 is 40.128403237855366

(d) The MSE for training set is 21.50447764071538
The MSE for test set is 25.59502793642659

1.1.5 Question 5

(a) We find the best σ and γ is:

$$\langle \sigma = 2^{8.5}, \gamma = 2^{-26} \rangle$$

(b) Plot the mean value over folds of validation error versus σ and γ, see Figure 10

(c) For the best σ and γ:

Best test error is 11.244792368852762 Best train error is 8.480217845690902
Best test error is 14.36446954565842 Best train error is 7.629959818826118

(d) see table 1

2 Part2

2.1 k-Nearest Neighbors

2.1.1 Question 6

Generate a dataset of 100 points with the coordinate of $[0, 1]^2$: X and 100 dummy targets of $[0, 1]$ to represent the label to training points: y to train the K-NN model. We set k = 3 by default, which means that for any input point, the nearest 3 points' majority label will be the label of that point. Then we can plot figure 11, Where red points are labeled with 1 and green points are labeled with 0. Any incoming points laid in the red zone will be labeled 1 and any incoming points laid in the green zone will be labeled 0.

Method	MSE train	MSE test
Native Regression	85.17 ± 4	83.25 ± 9
Linear Regression (attribute 1)	69.96 ± 4	76.74 ± 10
Linear Regression (attribute 2)	71.27 ± 4	78.21 ± 9
Linear Regression (attribute 3)	63.59 ± 5	67.22 ± 10
Linear Regression (attribute 4)	80.64 ± 4	84.99 ± 8
Linear Regression (attribute 5)	67.22 ± 5	72.93 ± 9
Linear Regression (attribute 6)	44.68 ± 4	41.91 ± 8
Linear Regression (attribute 7)	70.57 ± 5	76.48 ± 10
Linear Regression (attribute 8)	77.40 ± 5	83.11 ± 10
Linear Regression (attribute 9)	70.44 ± 4	75.91 ± 9
Linear Regression (attribute 10)	64.64 ± 4	68.76 ± 9
Linear Regression (attribute 11)	62.11 ± 4	64.17 ± 8
Linear Regression (attribute 12)	37.57 ± 3	40.59 ± 5
Linear Regression (all attributes)	22.71 ± 1	23.25 ± 4
Kernel Ridge Regression	7.30 ± 1	13.23 ± 2

Table 1: table for MSE vs regression method

2.1.2 Question 7

Calculate generalization error with:

$$error = \frac{1}{N}\sum_{n=1}^{N} I(y(x_n) \neq y_n)$$

Plot k against generalization error in Figure 12

1. When k is very low, the data is tending to select label only depending on a few points near it, which lead to overfitting to the training dataset which means that the KNN model will perform perfectly on training data but will generate a huge error on predicting new dataset.
2. When k gets large, it led to underfitting to predict new dataset which also led to increased generalization error.
3. And also, when k is even if the number of k nearest point labels are the same, the KNN algorithm tends to randomly select 1 or 0, hence the error of even ks is higher than odd ks

2.1.3 Question 8

Plot the graph of dataset size m against k-number k in Figure 13

With the increase of training points, the density of points increased because all points are laid in a fixed range, a low k number more easily causes overfitting. Hence with the increase in dataset size, the k number should also increase to avoid overfitting to give the lowest generalization error.

3 Part3

3.1 Question 9

(a) By theorem, K is positive semidefinite if and only if $K(x,t) = \langle \phi(x), \phi(t) \rangle$, where $x, t \in R^n$. Thus for

$$K_c(x,z) = C + \sum_{i=1}^{n} x_i z_i = \begin{bmatrix} \sqrt{c} \\ x_1 \\ x_2 \\ .. \\ .. \end{bmatrix}^T \begin{bmatrix} \sqrt{c} \\ z_1 \\ z_2 \\ .. \\ .. \end{bmatrix} = \langle \phi(x), \phi(z) \rangle$$

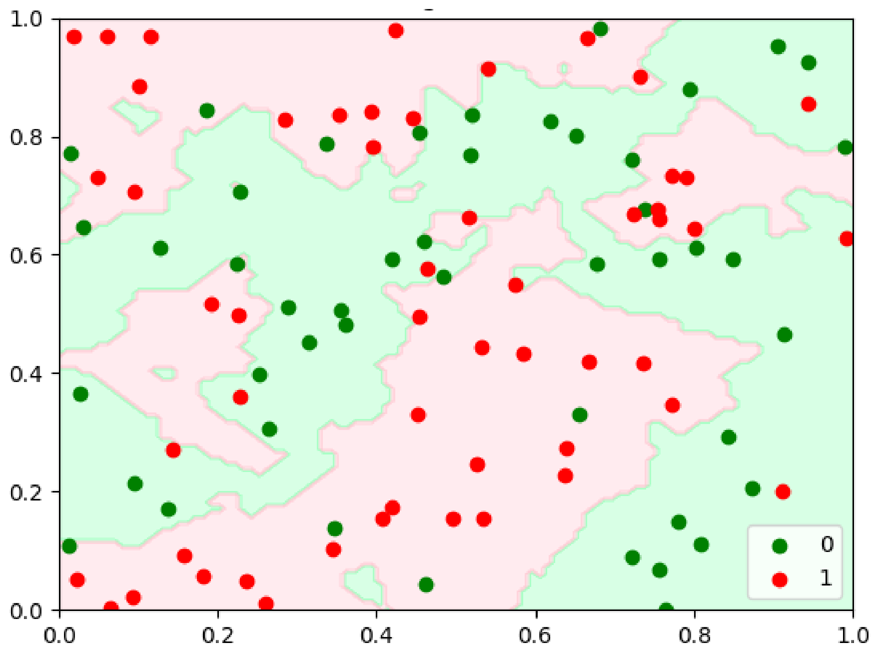

Figure 11: Visualisation decision boundary of trained KNN model

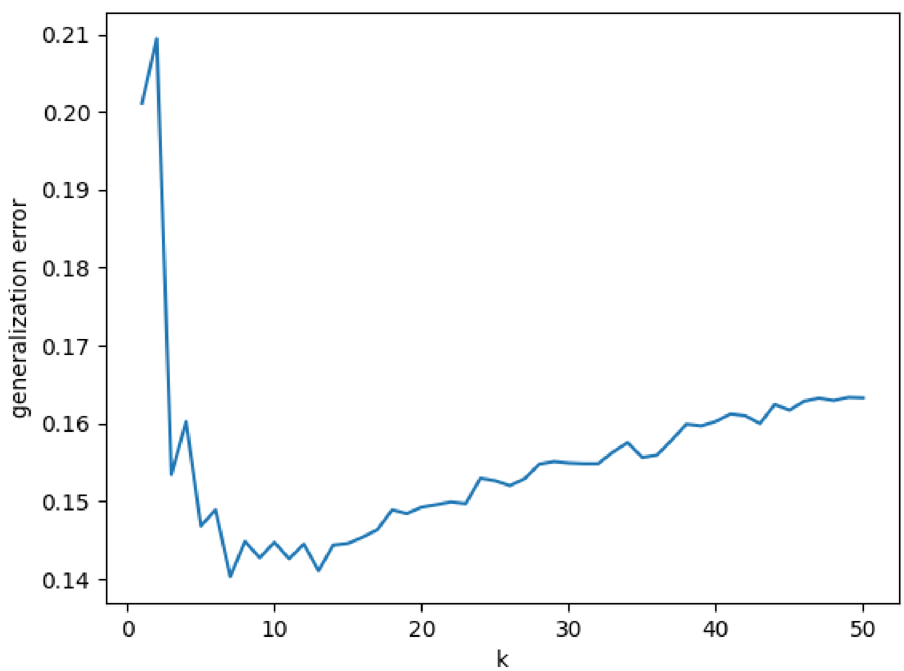

Figure 12: Plot k number against generalization error

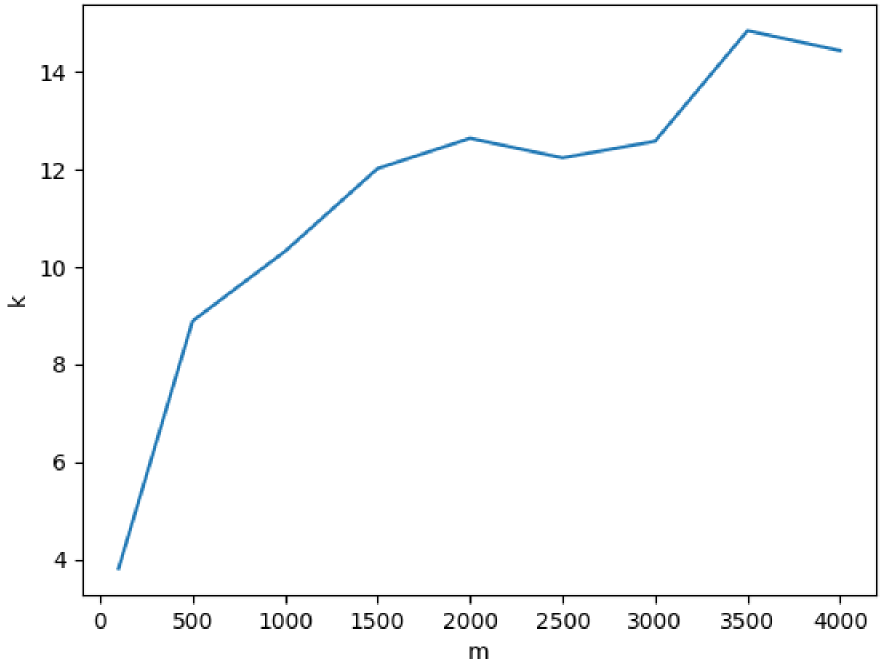

Figure 13: Plot optimal k value with dataset size

, where $\phi : R^n \to R^N$ is finite-dimensional feature map, and N = n+1. Then we need $\sqrt{C} \in R$, Thus $C \in [0, \infty]$.

(b) If we use K_c as a kernel function with linear regression. We got the feature map $\phi(x) = [\sqrt{c}, x_1, x_2, \cdot\cdot\cdot, x_m]$. Which just adds a bias term $\sqrt{C}$ to normal linear regression data set. Then as a bias term, $\sqrt{C}$ allows us to learn a fit with a constant offset and make regression function more away from the origin, thus allowing models to build relationships in non-zero centered data, which can make the solution converge better and accurately by changing the value of C.

3.2 Question 10

We need $\beta \to \infty$, which means β should be big enough to simulate a 1-NN classifier. That is because when $\beta \to \infty$, we have

$$K_\beta(x_i, x_j) = exp(-\beta \left|x_i - x_j\right|^2) \to 0$$

Thus the entries in K_β is quite small compared to I_m, then we have

$$\alpha = (K_\beta + \lambda I_m)^{-1} y \approx \frac{1}{\lambda} I_m y = \frac{1}{\lambda} y$$

where $\lambda > 0$. Then for

$$f(t) = \sum_{i=1}^{m} \alpha_i K_\beta(x_i, t) = \frac{1}{\lambda} \sum_{i=1}^{m} y_i exp(-\beta \left|x_i - t\right|^2)$$

where $exp(-\beta \left|x_i - t\right|^2) \to 0$, when x_i is far away from t, so $sign(y_i)$ will not affect the solution of $sign(f(t))$. But when x_i and t is close enough, then $\left|x_i - t\right|^2$ converges to 0 more faster than other points. Thus for the nearest point x_c, where $\left|x_c - t\right| = min\left|x_i - t\right|$ for $i \in [1, m]$, we have

$$exp(-\beta \left|x_c - t\right|^2) >> exp(-\beta \left|x_i - t\right|^2)$$

which means the nearest point have more weight to others if β is large. Hence

$$f(t) \approx \frac{1}{\lambda} y_c exp(-\beta \left| x_c - t \right|^2)$$

and

$$exp(-\beta \left| x_c - t \right|^2) > 0$$

Then $sign(f(t)) = sign(y_c)$, which means we simulate a 1-NN classifier here.