Introduction to ML (CS771), Autumn 2023
Indian Institute of Technology Kanpur
Homework Assignment Number 2

Student Name: Akash Shivaji Varude
Roll Number: 231110006
Date: November 15, 2023

QUESTION 1

- **Step 1 can be solved in following way**

  1. First, we have to calculate distance between all cluster centroids μ_k and data point x_n. For each cluster k, we have to calculate Euclidean distance between x_n and μ_k which is given by following formula:

  $$\text{distance}_{(x_n,\mu_k)} = \|x_n - \mu_k\|^2$$

  2. After this we have to assign data point x_n to cluster with the closest centroid. For this, find the cluster k with the smallest distance to x_n. Which is selecting cluster which minimizes the distance

  $$\arg\min_k \text{distance}_{(x_n,\mu_k)}$$

  3. Now just update data point x_n assignment. For this just set $z_{nk} = 1$ for cluster k that x_n was assigned to and 0 otherwise.

  Hence in alternate SGD iteration, update for z_n of data point x_n can be given by

  $$\hat{z}_n = \arg\min_{z_n} \sum_{k=1}^{K} z_{nk}\|x_n - \mu_k\|^2$$

  We can see, this equation is doing exactly same thing that we explained in above three steps

- **SGD-based cluster mean update equations for step 2**

  As K-means objective function is

  $$L = \sum_{n=1}^{N}\sum_{k=1}^{K} z_{nk}\|x_n - \mu_k\|^2$$

  The gradient of the loss with respect to the cluster mean μ_k is:

  $$\nabla_{\mu_k} L = -2\sum_{n=1}^{N} z_{nk}(x_n - \mu_k)$$

  Hence SGD based cluster means update equation is:

  $$\mu_k^{\text{new}} = \mu_k^{\text{old}} - \nabla_{\mu_k} L$$

  $$\mu_k^{\text{new}} = \mu_k^{\text{old}} + 2\eta\sum_{n=1}^{N} z_{nk}(x_n - \mu_k)$$

This Update step makes sense because

1. **Independence of Data Points**: In K-means, each data point's assignment to a cluster is independent of other data points. This property allows for individual updates. So adjusting cluster means based on a single data point's assignment is a valid approach.
2. **Efficiency and Scalability**: Updating one data point at a time is efficient and scalable, particularly for large datasets. It doesn't require processing the entire dataset in each iteration, hence it is suitable for online scenarios.
3. **Unsupervised learning nature**: K-means is unsupervised learning algorithm which is intented to find cluster patterns and does not require target values or labels.

- **Good choice of step size**

  We can take step size as $\frac{1}{N_k}$, where N_k is the number of data points in the k-th cluster.

  Reason: K-means can be sensitive to the initial cluster center positions. Using step size as $\frac{1}{N_k}$, would make updated mean in ratio of the sum of features of every data point to the total number of data points in that cluster. Also, as we are processing data in online manner, this step size simplifies the learning process.

Introduction to ML (CS771), Autumn 2023
Indian Institute of Technology Kanpur
Homework Assignment Number 2

Student Name: Akash Shivaji Varude
Roll Number: 231110006
Date: November 15, 2023

QUESTION 2

As we have to project data into one dimension such that maximizing the separation between class means and minimizing within-class variance are our goals, we can use Linear Discriminant Analysis (LDA). So, objective loss function to solve our problem is

$$L(w) = \frac{w^T S_b w}{w^T S_w w} \tag{1}$$

where

$L(w)$ is loss function to be maximized

w is the projection vector in $\mathbb{R}^D$

S_b is the between-class scatter matrix, and S_w is the within-class variance matrix, both defined based on class means and data distribution.

$$S_b = (\mu_1 - \mu_2)(\mu_1 - \mu_2)^T$$

where μ_1 and μ_2 are means of class +1 and class -1 respectively.

$$S_w = \sum_c \sum_n (x_n - \mu_c)(x_n - \mu_c)^T$$

for each class c and μ_c is mean of class c

Hence using loss function in equation (1), our objective function becomes,

$$f(w) = \arg\max_w \frac{w^T S_b w}{w^T S_w w} \tag{2}$$

Maximizing objective function in equation (2) will result in **maximizing** $S_b = (\mu_1 - \mu_2)(\mu_1 - \mu_2)^T$ i.e. **separation between class means**. and as S_w is in denominator it will **minimize within-class variance** i.e. $S_w = \sum_c \sum_n (x_n - \mu_c)(x_n - \mu_c)^T$

Justification for using this objective function

1. **Between-Class Scatter Matrix (S_b):**

   Maximizing Class Separation The term $(\mu_1 - \mu_2)(\mu_1 - \mu_2)^T$ in S_b quantifies the difference between the class means. So, by maximizing this term we ensure that the means of different classes are as far apart as possible in the projected space

2. **Within-Class Scatter Matrix (S_w):**

   Minimizing Within-Class Variance: The term $\sum_c \sum_n (x_n - \mu_c)(x_n - \mu_c)^T$ in S_w captures the spread of data points within each class. By minimizing this term, we ensure that data points within the same class are as close as possible in the projected space, which reduces overlap and improving classification accuracy

Introduction to ML (CS771), Autumn 2023
Indian Institute of Technology Kanpur
Homework Assignment Number 2

Student Name: Akash Shivaji Varude
Roll Number: 231110006
Date: November 15, 2023

QUESTION 3

As $\boldsymbol{v}$ is eigen vector of matrix $\frac{1}{N}\boldsymbol{X}\boldsymbol{X}^T$ we can write

$$\frac{1}{N}\boldsymbol{X}\boldsymbol{X}^T\boldsymbol{v} = \lambda\boldsymbol{v} \tag{1}$$

premultiplying equation (1) by $\boldsymbol{X}^T$ on both sides we get

$$\frac{1}{N}\boldsymbol{X}^T\boldsymbol{X}\boldsymbol{X}^T\boldsymbol{v} = \lambda\boldsymbol{X}^T\boldsymbol{v} \tag{2}$$

as we have covarience matrix $\boldsymbol{S} = \frac{1}{N}\boldsymbol{X}^T\boldsymbol{X}$, substituting this in equation (2), we get

$$\boldsymbol{S}\boldsymbol{X}^T\boldsymbol{v} = \lambda\boldsymbol{X}^T\boldsymbol{v} \tag{3}$$

so $\boldsymbol{X}^T\boldsymbol{v}$ is nothing but eigen vector of $\boldsymbol{S}$. But as given in question $\boldsymbol{u}$ is eigen vector of $\boldsymbol{S}$. **Hence, $\boldsymbol{u} = \boldsymbol{X}^T\boldsymbol{v}$ is eigen vector of $\boldsymbol{S}$**

Advantage of using this method: In the normal case, to compute K eigenvectors for $\frac{1}{N}(\boldsymbol{X}^T\boldsymbol{X})$, the complexity will be $O(KD^2)$. However, in this case, the complexity will be $O(KN^2) + O(KND)$ for decomposition and multiplication, respectively.

Therefore, the overall complexity will be $O(KND)$, which is less than $O(KD^2)$ as $N < D$.

Introduction to ML (CS771), Autumn 2023
Indian Institute of Technology Kanpur
Homework Assignment Number 2

Student Name: Akash Shivaji Varude
Roll Number: 231110006
Date: November 15, 2023

QUESTION 4

Part (1): Model assigns each data points to clusters, where each cluster is associated with a distinct weight vector $\boldsymbol{w_k}$ and then prediction is made for y by using corresponding $\boldsymbol{w_k}$. This allows the model to adapt to variations in the data, accommodating different relationships between inputs and outputs in various clusters. So the model can work in better way for complexities of heterogeneous data. While standard linear model with a single weight vector, which assumes a uniform relationship across all data points, our model allows for more accurate modeling by adapting to the variations within each cluster.

Part (2):
Conditional Posterior for our latent variable is

$$p(z_n = k|y_n, \Theta) = \frac{p(z_n = k)p(y_n|z_n = k, \Theta)}{\sum_{l=1}^{K} p(z_n = l)p(y_n|z_n = l, \Theta)} \tag{1}$$

where, we are given $p(z_n = k) = \pi_k$ and $p(y_n|z_n, \Theta) = \mathcal{N}(y_n|\boldsymbol{w}_{z_n}^\top \boldsymbol{x}_n, \beta^{-1})$
Hence equation (1) becomes

$$p(z_n = k|y_n, \Theta) = \frac{\pi_k \mathcal{N}(y_n|\boldsymbol{w}_k^\top \boldsymbol{x}_n, \beta^{-1})}{\sum_{l=1}^{K} \pi_l \mathcal{N}(y_n|\boldsymbol{w}_l^\top \boldsymbol{x}_n, \beta^{-1})}$$

$$p(z_n = k|y_n, \Theta) = \frac{\pi_k exp(\frac{-\beta}{2}(y_n - \boldsymbol{w}_k^\top \boldsymbol{x_n}))}{\sum_{l=1}^{K} \pi_l exp(\frac{-\beta}{2}(y_n - \boldsymbol{w}_l^\top x_n))} \tag{2}$$

ALT-OPT algorithm

- initialize $\Theta = \{\pi_k, \boldsymbol{w}_k\}$ as $\widehat{\Theta}$
- repeat until convergence
  For each n, compute most probable value (our best guess) of
  - Step 1: For each n, compute most probable value (our best guess) of z_n as

$$\widehat{z}_n = \arg\max_k p(z_n = k|y_n, \widehat{\Theta})$$

$$\Rightarrow \widehat{z}_n = \arg\max_k \frac{\widehat{\pi}_k exp(\frac{-\beta}{2}(y_n - \widehat{\boldsymbol{w}}_k^\top \boldsymbol{x_n}))}{\sum_{l=1}^{K} \widehat{\pi}_l exp(\frac{-\beta}{2}(y_n - \widehat{\boldsymbol{w}}_l^\top \boldsymbol{x_n}))}$$

– Step 2: Re-estimate the parameters

$$N_k = \sum_{n=1}^{N} \mathbb{1}_{z_n=k}$$

$$\boldsymbol{w}_k = (\boldsymbol{X}_k^T \cdot \boldsymbol{X}_k + \beta \boldsymbol{I})^{-1} \cdot \boldsymbol{X}_k^T \boldsymbol{y}_k$$

$$\pi_k = \frac{N_k}{N}$$

where $\boldsymbol{X_k}$ is $N_k \times D$ matrix which contain N_k data points of cluster k and $\boldsymbol{y}_k$ is $N_k \times 1$ vector of labels of inputs belonging to cluster k

Special Case: if $\pi_k = 1/K$ then

$$z_n = \arg\max_k \frac{exp(\frac{-\beta}{2}(y_n - \boldsymbol{w}_k^\top \boldsymbol{x_n}))}{\sum_{l=1}^{K} exp(\frac{-\beta}{2}(y_n - \boldsymbol{w}_l^\top x_n))} \tag{3}$$

We can see z_n will be assigned value using Multi-class Logistic Regression

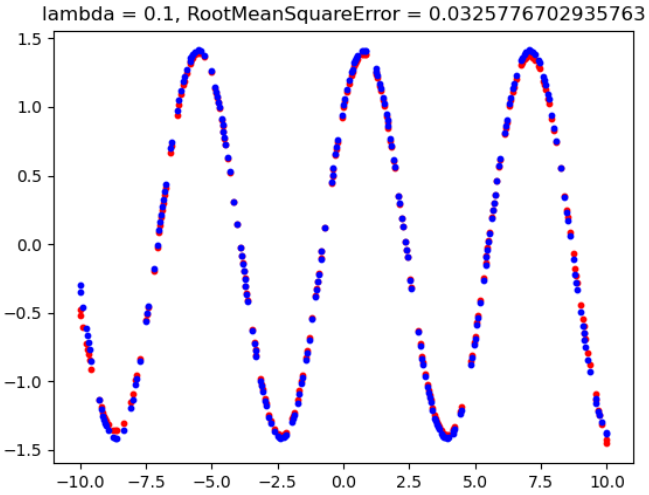
Introduction to ML (CS771), Autumn 2023
Indian Institute of Technology Kanpur
Homework Assignment Number 2

Student Name: Akash Shivaji Varude
Roll Number: 231110006
Date: November 15, 2023

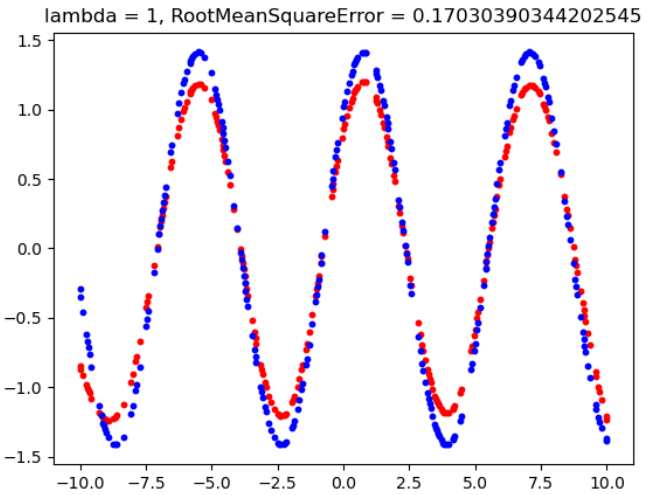
QUESTION
5

1. **Programming Problem Part 1:**

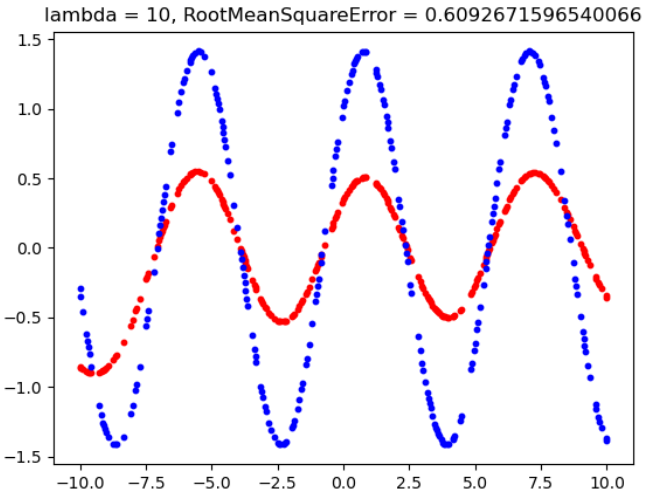
   (a) **kernel ridge regression:**
   for lambda = 0.1, RMS error = 0.0325776702935763



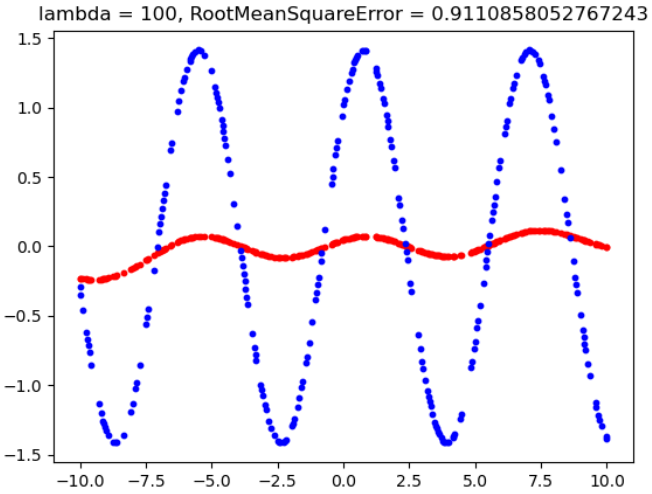
   for lambda = 1, RMS error = 0.17030390344202545



for lambda = 10, RMS error = 0.6092671596540066



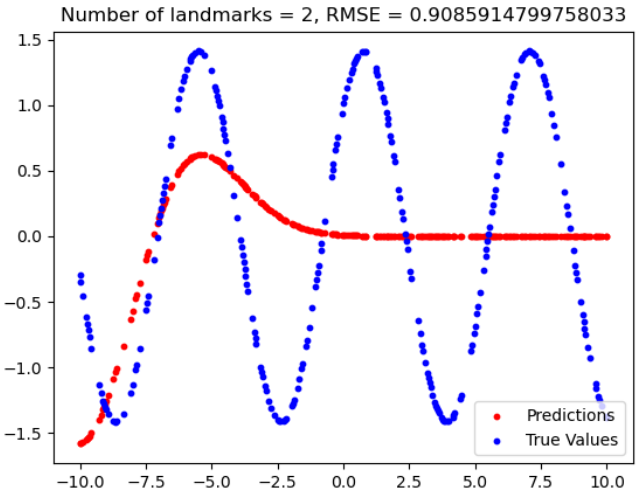
for lambda = 100, RMS error = 0.9110858052767243



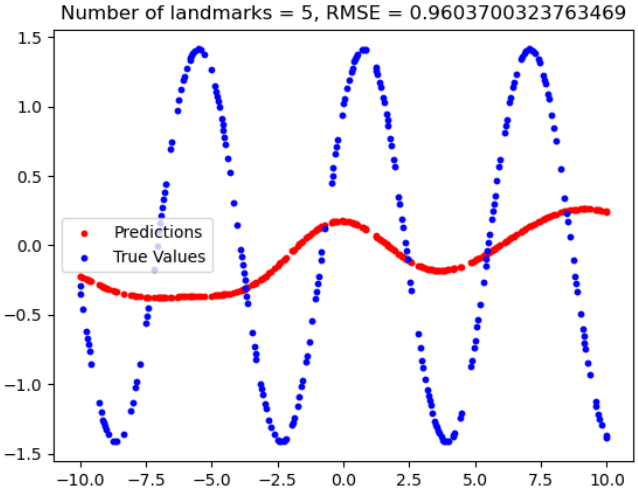
Observation: Increasing the regularization parameter (λ) from 0.1 to 100 leads to a gradual shift. As we can see with λ= 0.1, accuracy is greater than 99.97. So it will be good chouce for λ. However smaller value of λ than that may cause overfitting. As λ increases, the model becomes more constrained. However, with larger λ (e.g., 100), the model becomes highly constrained, leading to underfitting causing very high test error.

(b) **landmark-ridge Regression:**

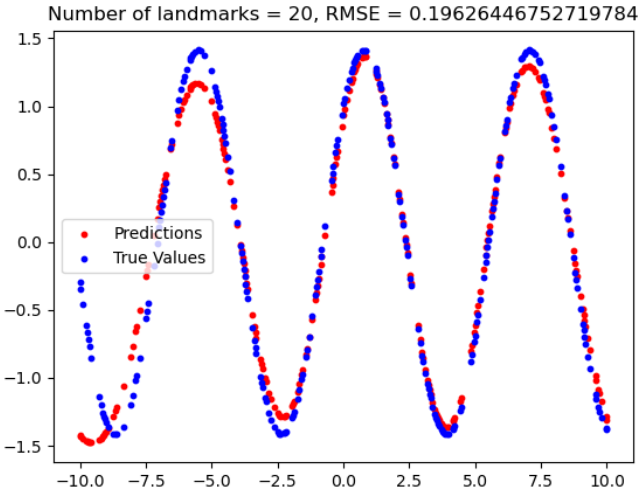
RMSE for number of 2 landmarks = 0.9085914799758033



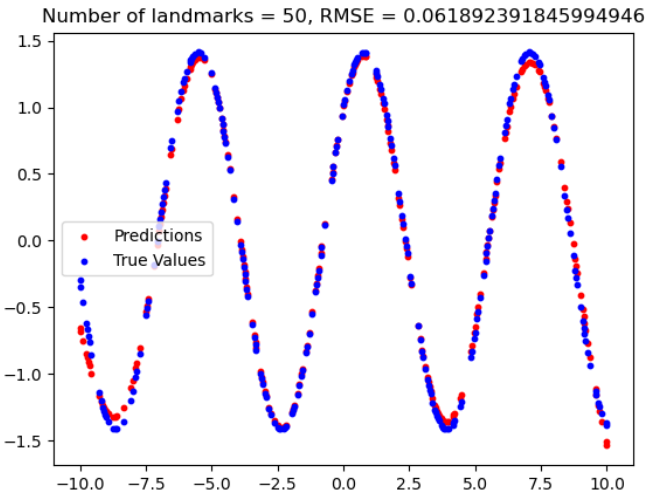
RMSE for number of 5 landmarks = 0.9603700323763469



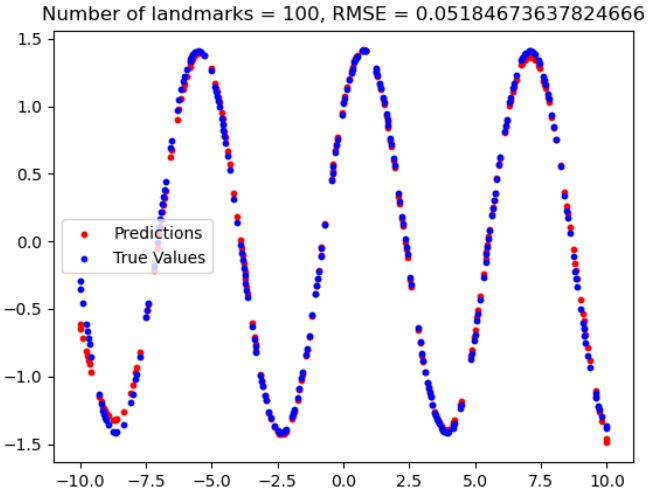
RMSE for number of 20 landmarks = 0.19626446752719784



RMSE for number of 50 landmarks = 0.061892391845994946



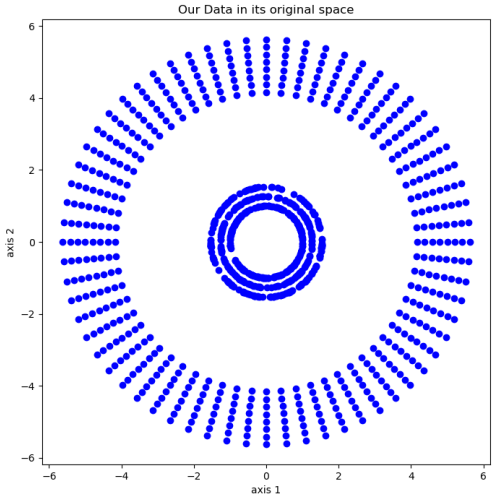
RMSE for number of 100 landmarks = 0.05184673637824666



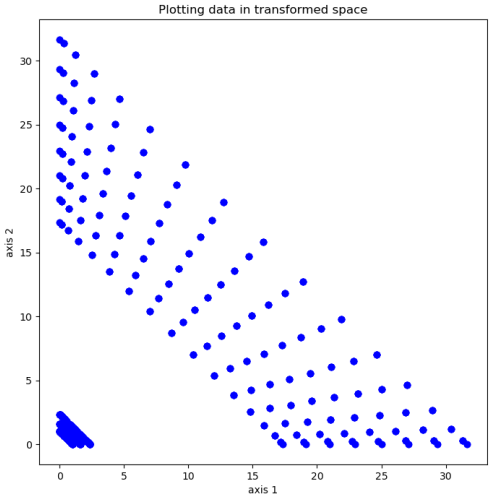
Observation: The RMSE values for different numbers of landmarks decreases the error as the number of landmarks increases. Hence more landmarks helps the model capture the underlying patterns in the data more effectively. The optimal number of landmarks appears to be around 50, As the RMSE reaches low value around this. Beyond this, the improvement in RMSE becomes marginal and model may become more complex which may cause overfitting.

2. **Programming Problem Part 2:**

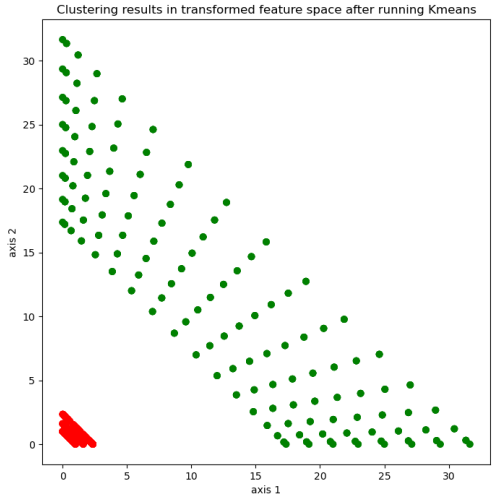
   (a) **K-means using handcrafted features:**



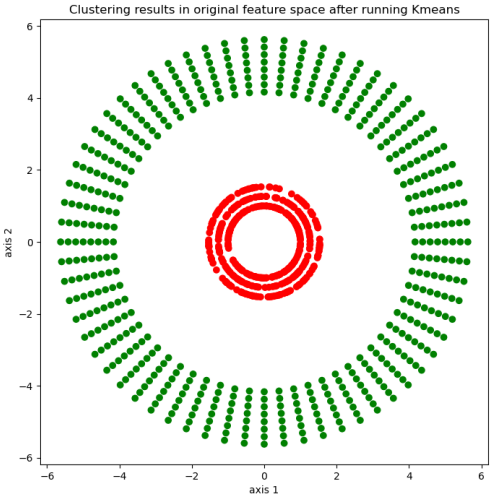
(a) Data in Original Space



(b) Data in Transformed Space

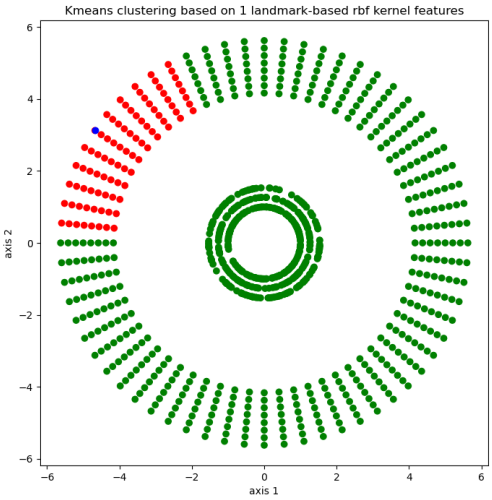


(c) Result in Transformed Space

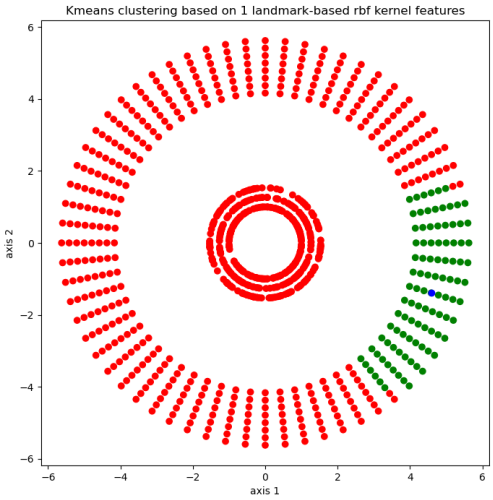


(d) Result in Original Space

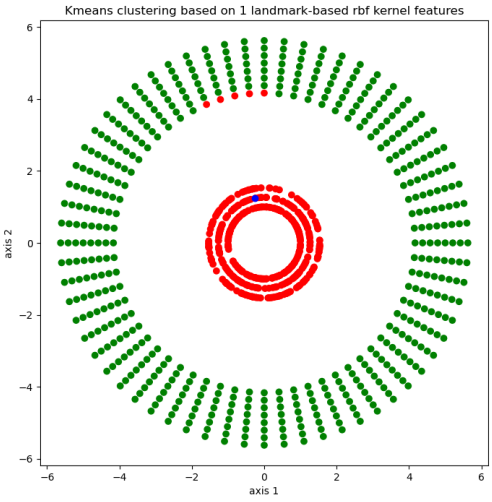
(b) **K-means using RBF kernel:**



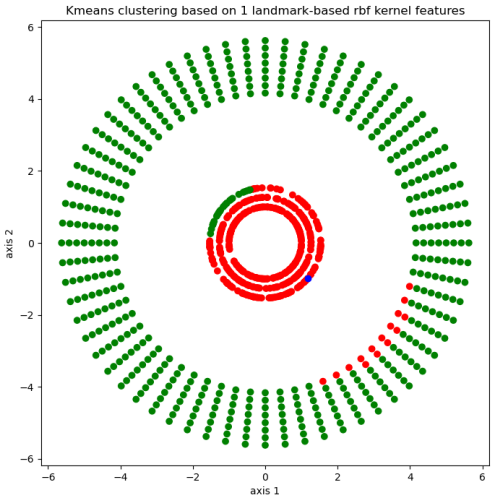
(a) Figure 1



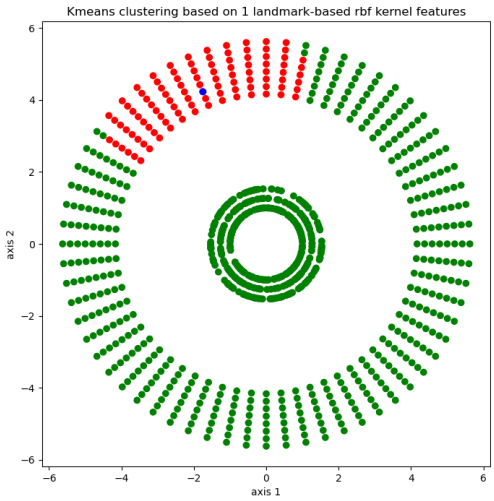
(b) Figure 2



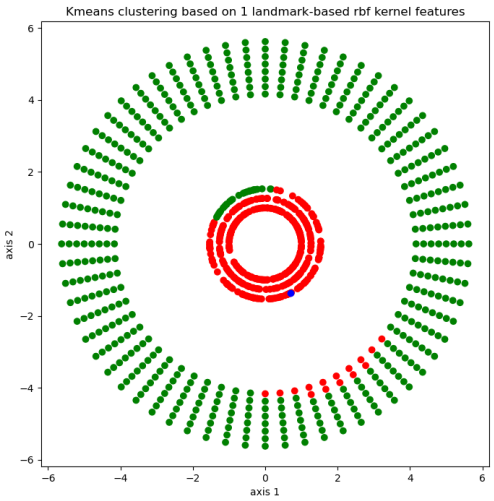
(c) Figure 3



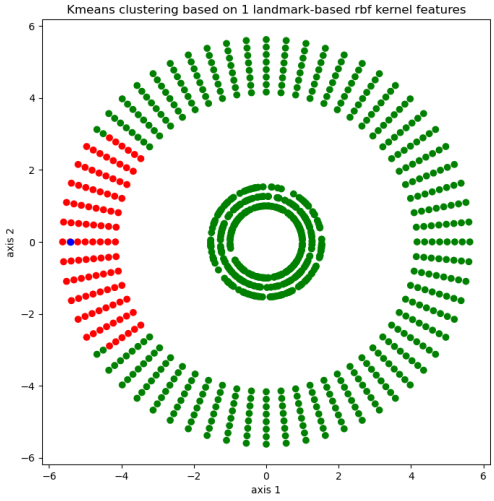
(d) Figure 4



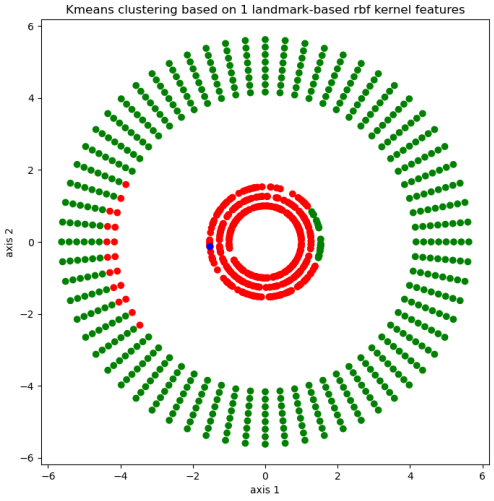
(a) Figure 5



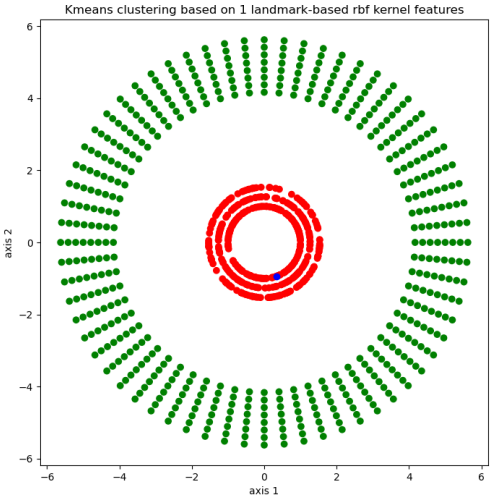
(b) Figure 6



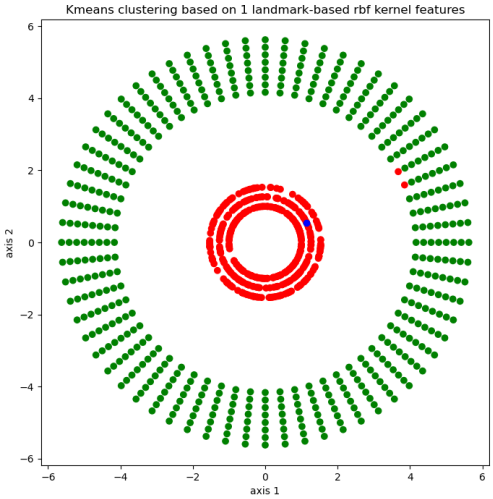
(c) Figure 7



(d) Figure 8

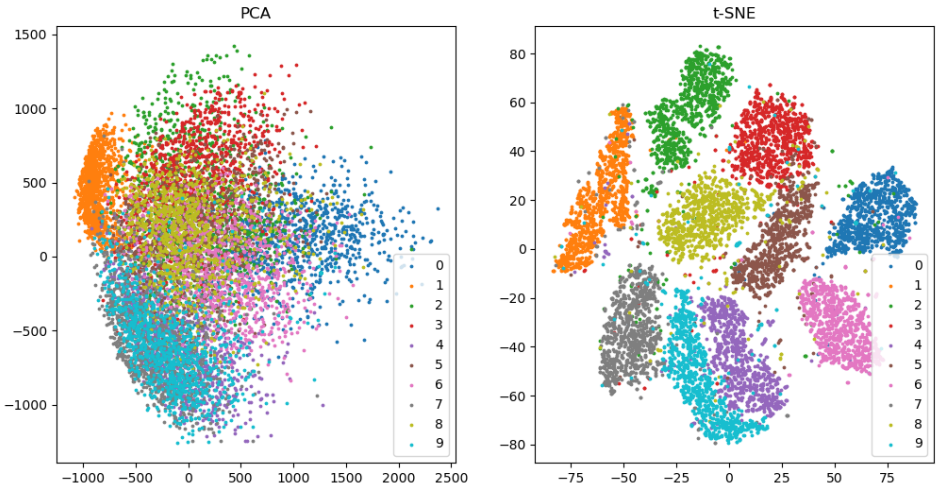


(a) Figure 9



(b) Figure 10

3. **Programming Problem Part 3: PCA and tSNE**



Observation:

While comparing above PCA and t-SNE plots, we can see, PCA stretches the data along its principal components, emphasizing the global structure. Due to which there is overlapping between points from different classes.

On the other hand, t-SNE is a non-linear method for dimensionality reduction method. Hence it grouped similar points together, revealing intricate local structures and clusters more explicitly.

So, in PCA plot, there is a spread-out representation, while t-SNE plot display clusters of similar points