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

Student Name: YASHVIR SINGH NATHAWAT
Roll Number: 231110059
Date: November 17, 2023

QUESTION 1

The original k-means loss function is given as:

$$L(X, Z, \mu) = \sum_{n=1}^{N} \sum_{k=1}^{K} z_{nk} \|x_n - \mu_k\|^2$$

In online version:

Initialize cluster centers $\mu_1, \mu_2, \dots, \mu_k \in \mathbb{R}^d$ randomly.
Set cluster count $n_1, n_2, \dots, n_k \in \mathbb{N}$ to 0.
PART A.)

In Step 1 online version we take random example x_n at a time, and then greedily assign x_n to the best cluster.

After fixing $\mu = \hat{\mu}$, find z_n. We are finding distance of x_n from every $\hat{\mu}$ and whichever distance is minimum , x_n is assigned to that cluster.

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

Increase the z_nth cluster count by 1

$$n_{zn} = n_{zn} + 1$$

PART B.)
STEP 2: Now we will update the cluster means

Now we will fix $z_n = \hat{z_n}$ then Solving for μ_k using SGD:

$$\hat{\mu} = \arg\min_{\mu} L(X, \hat{Z}, \mu)$$

$$\hat{\mu} = \arg\min_{\mu} \sum_{n=1}^{N} \sum_{n:\hat{z}_n = k} \|x_n - \mu_k\|^2$$

$$\hat{\mu_k} = \arg\min_{\mu_k} \sum_{n:\hat{z}_n = k} ||x_n - \mu_k||^2$$

As in SGD, at any iteration t we will choose x_n uniformly randomly and the approximate gradient g.

Stochastic Gradient based updates are :

$$\mu_k = \mu_k - \eta g_n$$

where g_n is the gradient of L with respect to μ_k.

$$g \approx g_n = \frac{\partial}{\partial \mu_k} \left(\|x_n - \mu_k\|^2 \right) = -2(x_n - \mu_k)$$

Now the update equation of mean in SGD is following:

$$\mu_k^{(t+1)} = \mu_k^{(t)} - \eta g^{(t)}$$

$$\mu_k^{(t+1)} = \mu_k^{(t)} + 2\eta(x_n^{(t)} - \mu_k^{(t)})$$

Intuitively, this update equation seems intuitively reasonable since it minimizes the overall objective function L by shifting the cluster mean μ_k in the direction of the data points assigned to cluster k. The negative gradient term pulls the cluster mean towards the weighted average of the data points assigned to the cluster.

PART C.)

The step size can be $\eta \propto \frac{1}{N_k}$, where N_k is the number of data points in the k-th cluster, so that the updated mean is in the ratio of the sum of features of every data point to the total number of data points in that cluster.

The learning rate η is inversely proportional to the number of data points N_k in the cluster, making this step size an excellent choice.This ensures that larger clusters contribute less to the update, preventing them from dominating the adjustment of the cluster mean. It facilitates the achievement of an even distribution of data point influences on the cluster mean updates.

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

Student Name: YASHVIR SINGH NATHAWAT
Roll Number: 231110059
Date: November 17, 2023

QUESTION 2

We are given some labeled training data $\{x_n, y_n\}$ with inputs $x_n \in \mathbb{R}^D$ and labels $y_n \in \{-1, +1\}$. We want to project the inputs into one dimension using a projection direction given by a vector $w \in \mathbb{R}^D$ such that, after the projection, the distance between the means of the inputs from the two classes becomes as large as possible, and the inputs within each class become as close to each other as possible. The objective function to achieve this is given by:

$$J = \frac{(m_1 - m_2)^2}{S_1^2 + S_2^2} = \frac{W^T S_B W}{W^T S_W W}$$

where S_1^2 and S_2^2 are the covariances matrices of the two classes, and m_1 and m_2 are the means of the two classes,W is the projection direction, S_B is the between-class scatter matrix, and S_W is the within-class scatter matrix.

Explanation :

The numerator here is **between class scatter** while the denominator is **within-class scatter**. Numerator is distance between the means of two classes, maximizing it will increase the distance between the means of inputs from the two classes as large as possible. Denominator is within class scatter or variance so minimizing it will make inputs within each class as close as possible to each other in same class.

So to maximize the function we need to maximize the numerator and minimize the denominator.

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

Student Name: YASHVIR SINGH NATHAWAT
Roll Number: 231110059
Date: November 17, 2023

QUESTION

3

For X being an $N \times D$ matrix:

$$X \in \mathbb{R}^{N \times D}$$

And for S being a $N \times N$ matrix:

$$S = \frac{1}{N} X X^T \in \mathbb{R}^{N \times N}$$

Given eigenvector $v \in \mathbb{R}^N$ of the matrix $\frac{1}{N} X X^T$.The eigenvector equation is:

$$S\mathbf{v} = \lambda \mathbf{v} \quad \text{where } \lambda \text{ is the eigenvalue.}$$

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

After multiplying each side by X^T:

$$X^T \left(\frac{1}{N} X X^T \mathbf{v} \right) = X^T \lambda \mathbf{v}$$

$$\frac{1}{N} X^T X (X^T \mathbf{v}) = \lambda (X^T \mathbf{v})$$

Let $p = X^T \mathbf{v}$. The equation becomes:

$$\frac{1}{N} (X^T X) p = \lambda \mathbf{p}$$

Therefore, $p = X^T \mathbf{v}$ is an eigenvector of the covariance matrix $\frac{1}{N} X^T X$ with the same eigenvalue λ.
So we can compute the eigenvectors of the covariance matrix $S = \frac{1}{N} X^T X$.

Advantage of this method:
Traditionally, computing the top k eigenvectors takes $\mathcal{O}(k \times d^2)$, where k is the number of top eigenvectors and d is the dimension of the input x^n
While in this method, For the decomposition of $\frac{1}{N} X X^T$, it takes $O(KN^2)$ time using SVD(Singular Value Decomposition).
Multiplication of matrices takes $O(KND)$.
Therefore, the overall time complexity is $O(KND)$, which is less than $O(KD^2)$ since $N \ll D$.
So when $N \ll D$ new method is **faster** than traditional method.

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

Student Name: YASHVIR SINGH NATHAWAT
Roll Number: 231110059
Date: November 17, 2023

QUESTION 4

Part (1)

In a standard probabilistic linear model, a single weight vector is used to capture the relationship between inputs and responses, assuming a linear curve. In contrast, the introduced model with latent variables and cluster-specific weight vectors allows for a more flexible representation and combines K different linear curves. By clustering the data into K different groups, each associated with its own linear curve, the model can capture more complex relationships and variations in the data and give prediction for y. This approach not only accommodates non-linearity but also provides a way to address outliers by potentially assigning them to different clusters with distinct linear patterns,helping to reduce their impact on the overall regression model.

Part (2)

The ALT-OPT algorithm aims to estimate the latent variables Z and the maximum likelihood estimates (MLE) of the global parameters $\Theta = \{(w_1, \ldots, w_K), (\pi_1, \ldots, \pi_K)\}$. The algorithm alternates between updating the latent variables and the global parameters in an iterative manner. Given

$$p(y_n|z_n, x_n, W) = \mathcal{N}(y_n|w_{z_n}^T x_n, \beta^{-1})$$

$$p(z_n) = \text{multinoulli}(z_n|\pi_1, \ldots, \pi_K) = \pi_k$$

Our latent variable model is the Posterior probability of the latent variable z_n taking the value k given the observed data y_n and the model parameters θ.

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

Ignoring the denominator term, we get

$$p(y_n, z_n|\theta) = p(y_n|z_n, \theta) \cdot p(z_n|\theta)$$

Using ALT-OPT algorithm :

Step 1 : Finding optimum z_n while keeping the other parameters as fixed::

$$z_n = \arg\max_{z_n} \frac{\pi_k \cdot \mathcal{N}(w_{z_n}^T x_n, \beta^{-1})}{\sum_{l=1}^{K} \pi_l \cdot \mathcal{N}(w_l^T x_n, \beta^{-1})}$$

Given ,

$$\mathcal{N}(w_{z_n}^T x_n, \beta^{-1}) = \exp\left(-\frac{\beta}{2}(y_n - w_{z_n}^T x_n)^2\right)$$

$$\implies z_n = \arg\max_{z_n} \frac{\pi_k \cdot \exp\left(-\frac{\beta}{2}(y_n - w_{z_n}^T x_n)^2\right)}{\sum_{l=1}^{K} \pi_l \cdot \exp\left(-\frac{\beta}{2}(y_n - w_l^T x_n)^2\right)}$$

Intuitively, the updates for z_n aim to find the cluster assignment that maximizes the probability of the observed data y_n under the given model. This involves comparing the likelihood of y_n under the current cluster k against the likelihoods for all clusters, weighted by their prior probabilities π_k, and selecting the cluster that maximizes this probability.

Step 2 : Fixing z_n and finding optimal value w_k, N_k and pi_k :

Taking log in Posterior probability equation we get,

$$\log p(y_n, z_n|\theta) = \log p(y_n|z_n, \theta) + \log p(z_n|\theta)$$

$$= \log(\pi_k) + \log(\cdot\mathcal{N}(w_{z_n}^T x_n, \beta^{-1}))$$

$$= \log(\pi_k) - \frac{\beta}{2}(y_n - w_{zn}^T x_n)^2$$

Differentiate with respect to w_k, we get

$$w_k = (X_k^T X_k)^{-1} X_k^T y_k$$

In this scenario, the training instances belonging to class k are organized into a matrix X_k with dimensions $N_k \times D$, where N_k is the number of instances in class k, and D is the feature dimensionality. Correspondingly, the labels of the training instances in class k are structured into vectors y_k with dimensions $N_k \times 1$.

Differentiate with respect to pi_k, we get

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

where,

$$N_k = \sum_{n=1}^{N} z_{nk}$$

Intuitively,The update equations for w_k, N_k, and π_k are estimating the parameters of the model. w_k represents the optimal weight vector for cluster k, N_k is the count of data points assigned to cluster k, and π_k is the proportion of data points belonging to cluster k in the entire dataset. These updates collectively refine the model parameters to better capture the underlying patterns in the data.

If $\pi_k = \frac{1}{K}$, then:

$$z_n = \arg\max_{z_n} \frac{\exp\left(-\frac{\beta}{2}(y_n - w_{zn}^T x_n)^2\right)}{\sum_{l=1}^{K} \exp\left(-\frac{\beta}{2}(y_n - w_l^T x_n)^2\right)}$$

This update is equivalent to softmax classification, overall is multi-output logistic regression.

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

Student Name: YASHVIR SINGH NATHAWAT
Roll Number: 231110059
Date: November 17, 2023

QUESTION

5

Programming Problem: Part 1

1.Kernal ridge regression:

RMSE - Observations

λ	RMSE
0.1	0.03257767029357466
1	0.17030390344202526
10	0.6092671596540067
100	0.9110858052767243

Our observation is that when we increase the regularization parameter, or λ, our RMSE value likewise increases. This indicates that if we increase the λ value, our model's prediction accuracy decreases.Given that the train and test sets seem to have been drawn from the same distribution set with few outliers,a better curve toward train data and, consequently, a better curve toward test data are achieved with less regularization.
Plots are following:

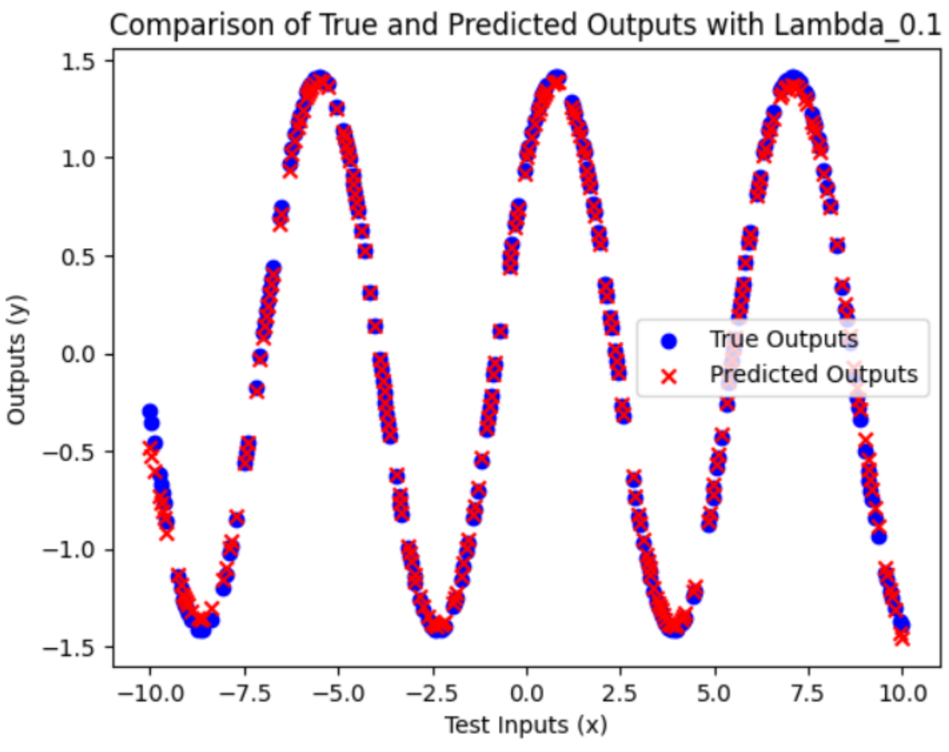
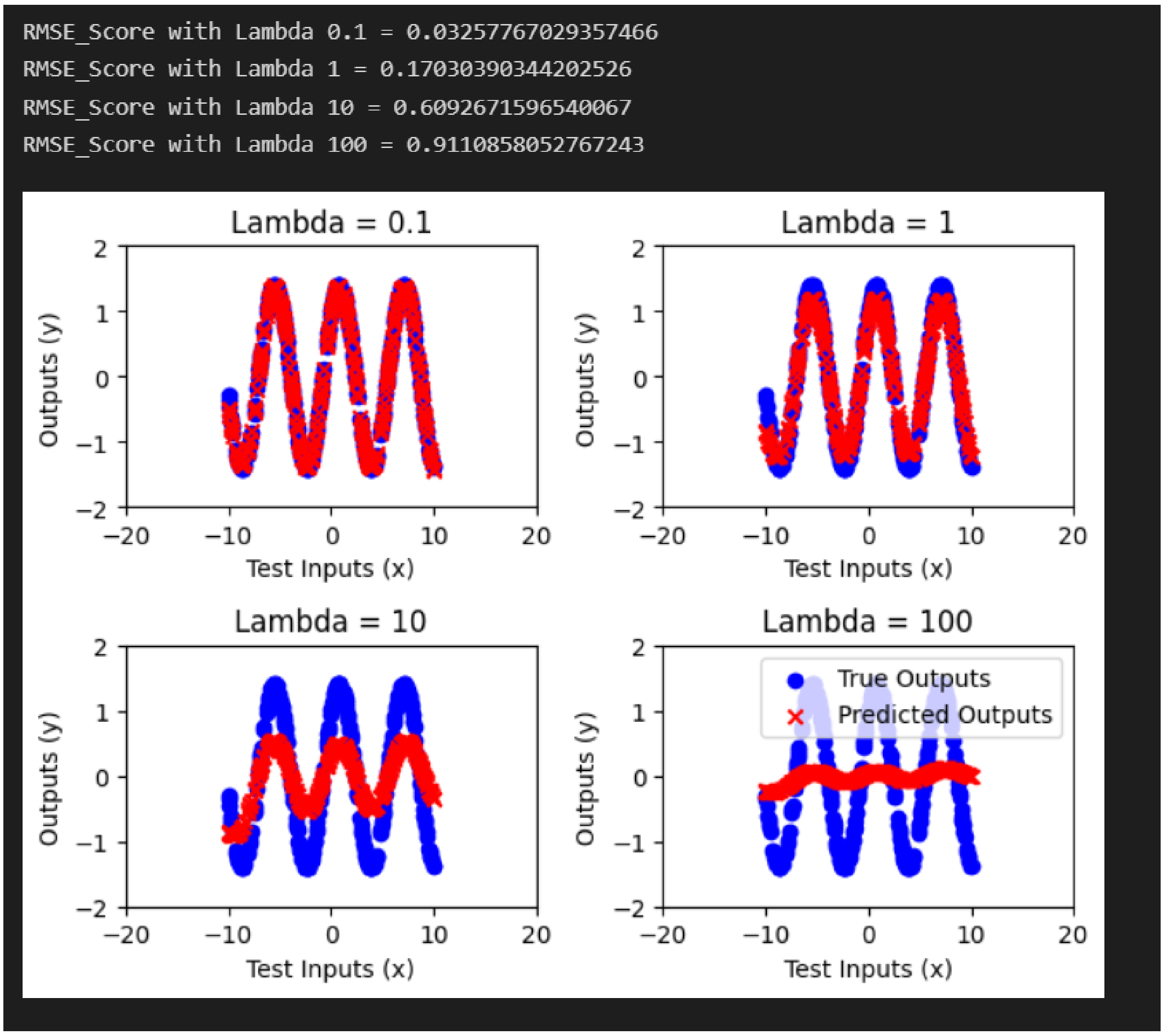


Figure 1: for λ=0.1



```
RMSE_Score with Lambda 0.1 = 0.03257767029357466
RMSE_Score with Lambda 1 = 0.17030390344202526
RMSE_Score with Lambda 10 = 0.6092671596540067
RMSE_Score with Lambda 100 = 0.9110858052767243
```

Figure 2: for λ=[0.1,1,10,20]

2.Landmark ridge regression:

While λ remains the same, the value of L is altered. Since we are selecting landmarks from the data set uniformly at random, the results of each program run may vary.

Observations : Prediction Error keeps on decreasing as we increase L number of landmarks. L = 50 is good choice as with L = 100 and L = 50 there is very less difference in RMSE-values.

RMSE - Observations
Fixed λ value is : 0.1

L	RMSE-Score
2	0.9734417342075632
5	0.9009978600999894
20	0.33125246200097785
50	0.08828022494192253
100	0.05249835877151603

Plots are following:

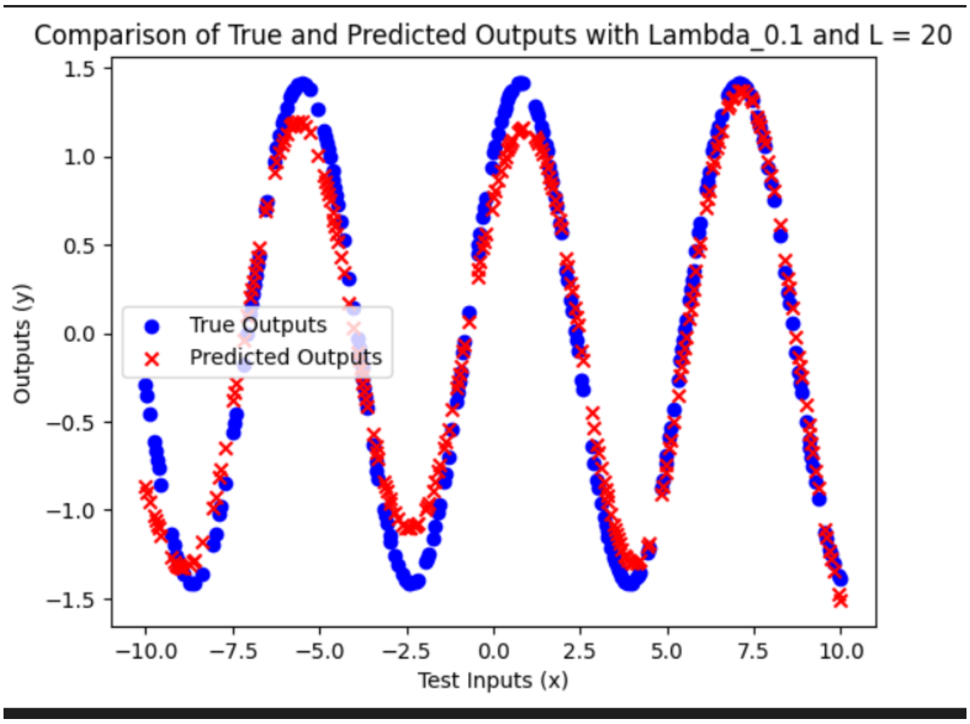
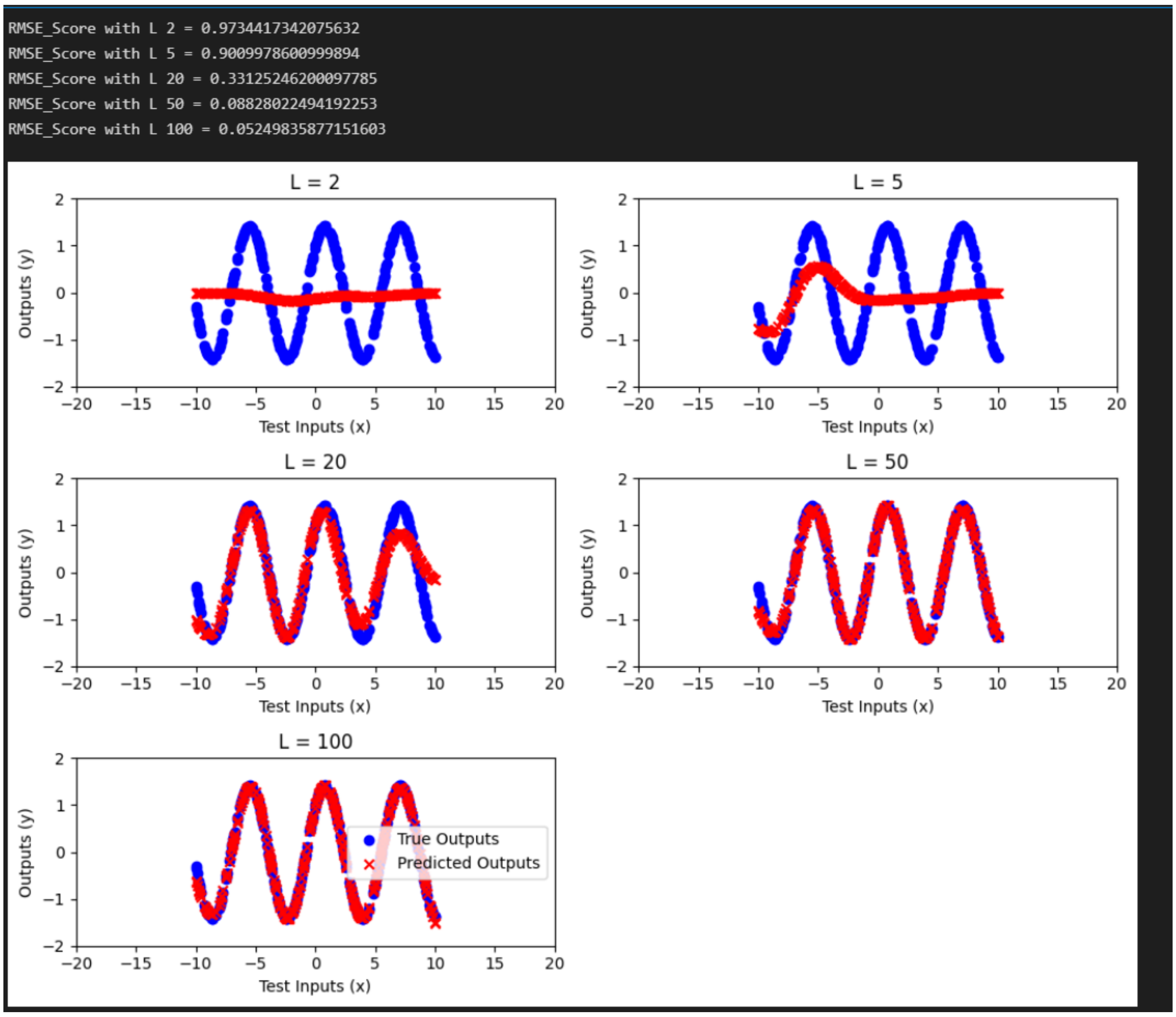


Figure 3: for L=20



```
RMSE_Score with L 2 = 0.9734417342075632
RMSE_Score with L 5 = 0.9009978600999894
RMSE_Score with L 20 = 0.33125246200097785
RMSE_Score with L 50 = 0.08828022494192253
RMSE_Score with L 100 = 0.05249835877151603
```

Figure 4: for L=[2,5,20,50,100]

Programming Problem: Part 2

1. Using Hand-crafted Features::
Original 2D K-Means Data [x,y] :

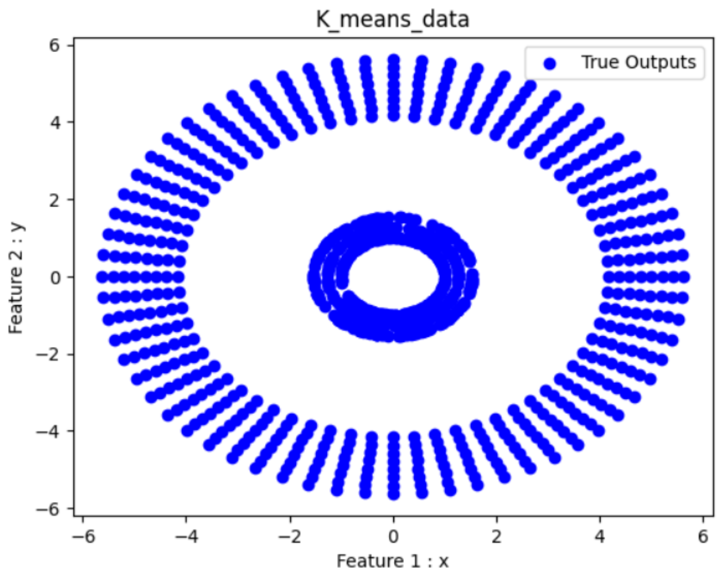


Figure 5: Original data before applying transformation

Transforming into 3D using Transformation function :

$$[z_1, z_2, z_3] \Rightarrow [x^2, \sqrt{2} \cdot |xy|, y^2]$$

Following the transformation, we run K-means on it and plot the clustering result in the original 2D space as shown below:

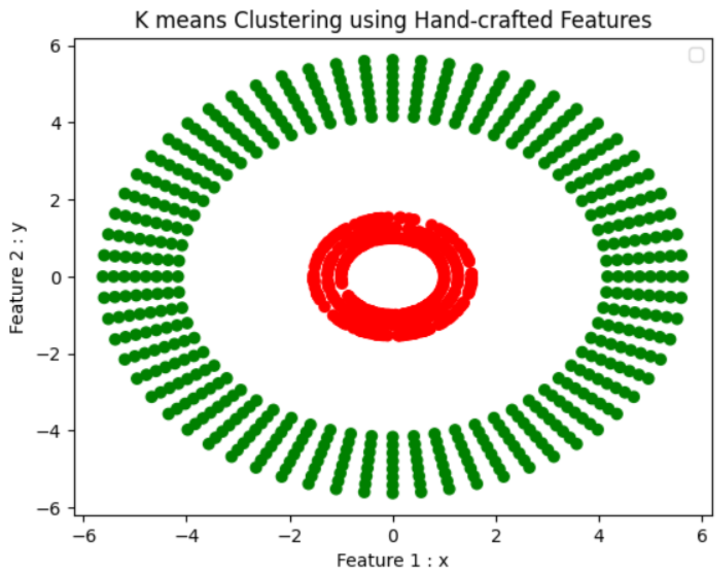


Figure 6: K-means result on original 2D data

2. Using Kernels::

Since we are uniformly selecting landmarks at random from the data set, the results of each program run could differ. If we apply a feature transform based on the distance of the point from the origin, we can easily cluster the given data. When a landmark is closer to the origin, it can sometimes cluster the data well; however, when the landmark is farther away from origin, the data are dispersed radially around the origin, so the data do not cluster well.
Following are the plots:
blue square is landmark point.
Note:-Landmarks : [-3.5429646 2.9076359] means landmarks with feature 1 and feature 2.

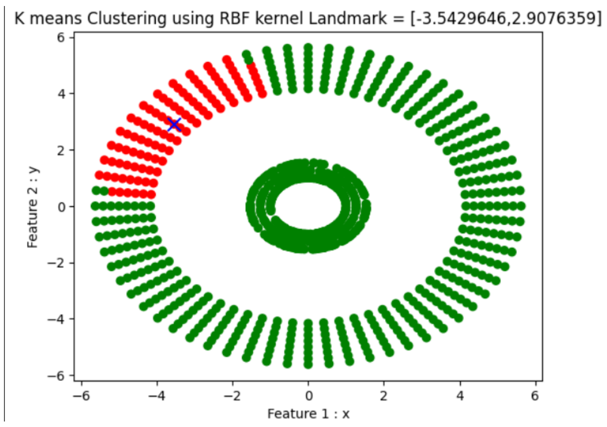


Figure 7: Landmarks : [-3.5429646 2.9076359]

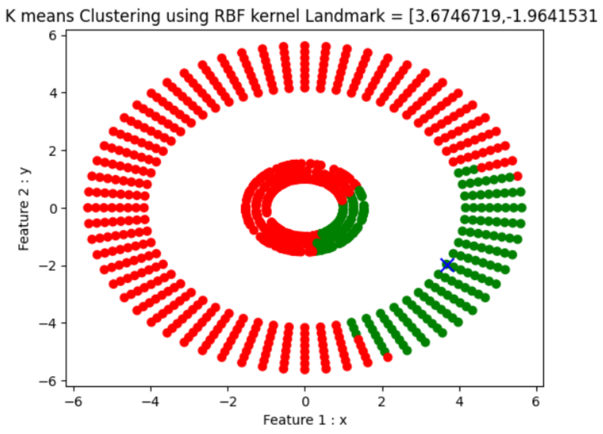


Figure 8: Landmarks : [3.6746719 -1.9641531]

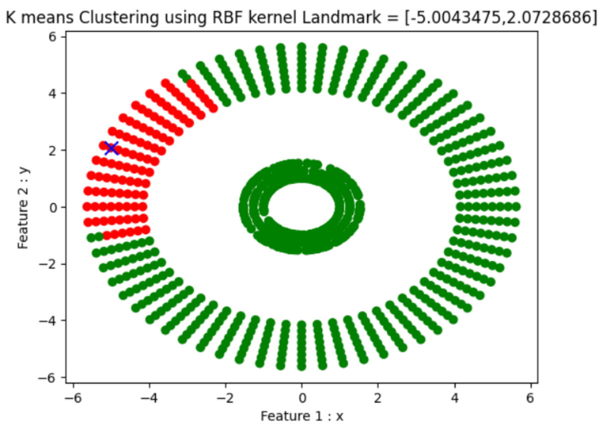


Figure 9: Landmarks : [-5.0043475 2.0728686]

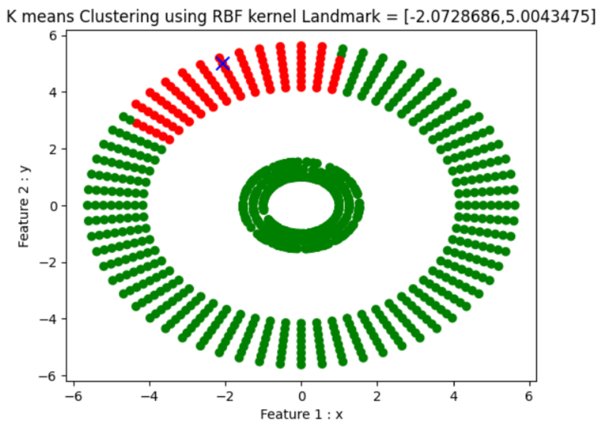


Figure 10: Landmarks : [-2.0728686 5.0043475]

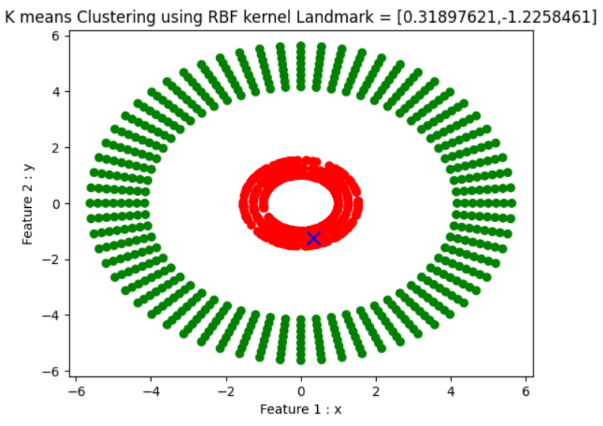


Figure 11: Landmarks : [0.31897621 -1.2258461]

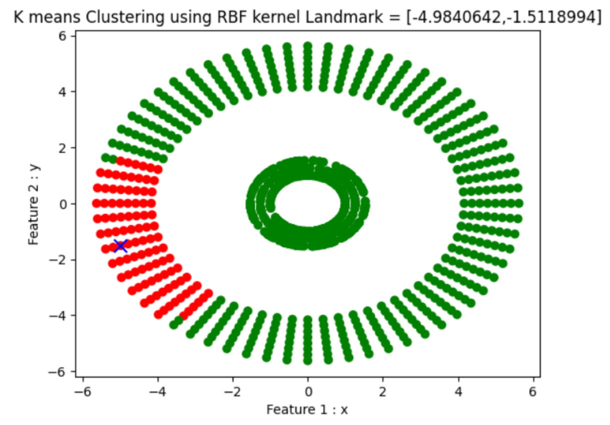


Figure 12: Landmarks : [-4.9840642 -1.5118994]

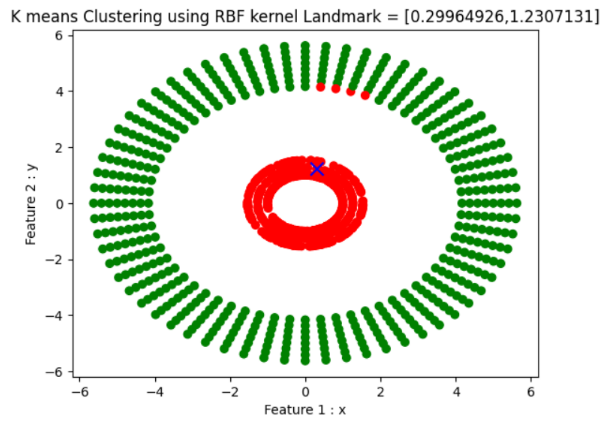


Figure 13: Landmarks : [0.29964926 1.2307131]

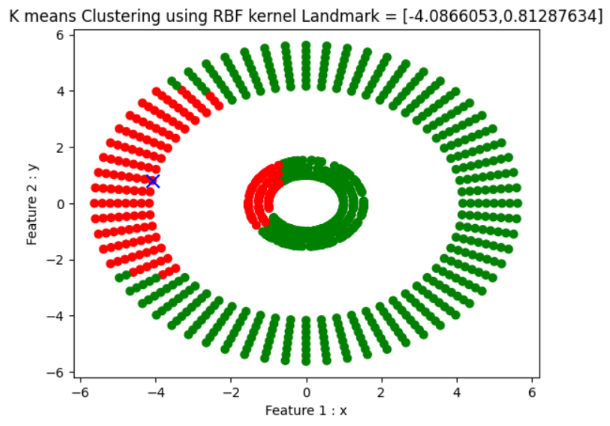


Figure 14: Landmarks : [-4.0866053 0.81287634]

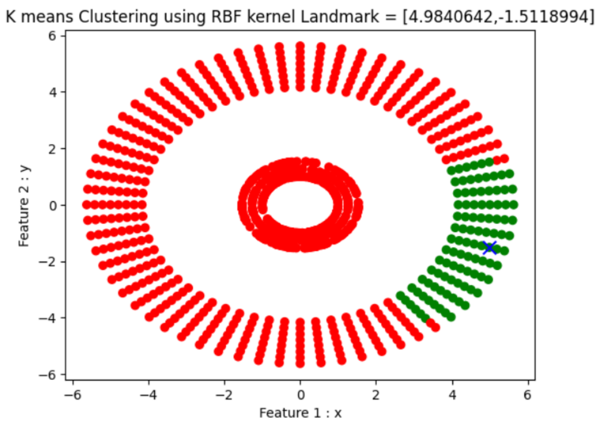


Figure 15: Landmarks : [4.9840642 -1.5118994]

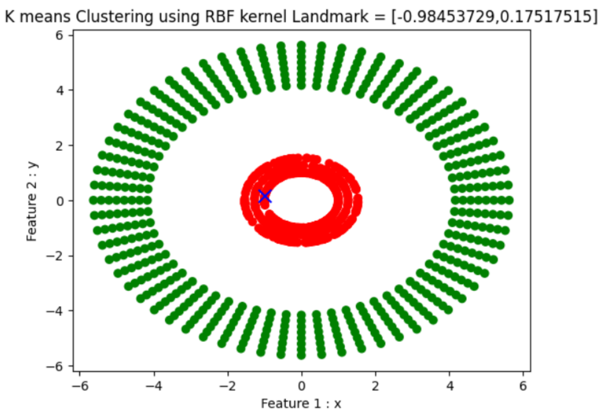


Figure 16: Landmarks : [-0.98453729 0.17517515]

Programming Problem: Part 3

Observation : t-SNE tends to be better at preserving local structures and capturing non-linear relationships between data points. In contrast, PCA focuses on maximizing variance along the principal components, which may not capture complex non-linear relationships as effectively as t-SNE. Therefore, t-SNE plots often reveal more intricate patterns in the data.Clusters in tSNE appear to be more separated visually then PCA, and errors in tSNE seem to be lower than those in PCA when ten different initializations are done.

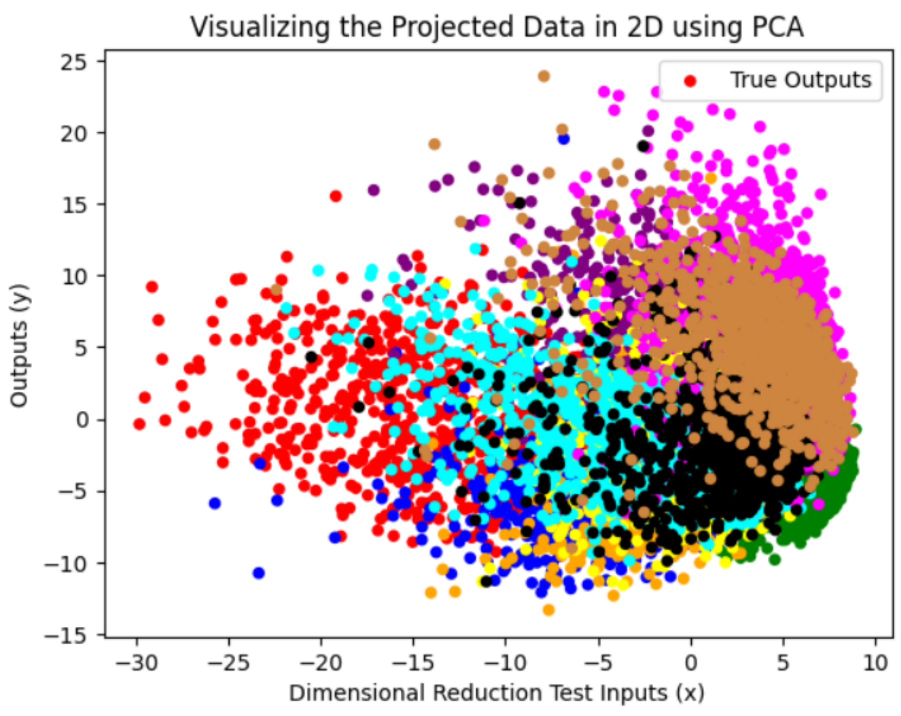


Figure 17: Visulaization the projected data in two dimensions using PCA method

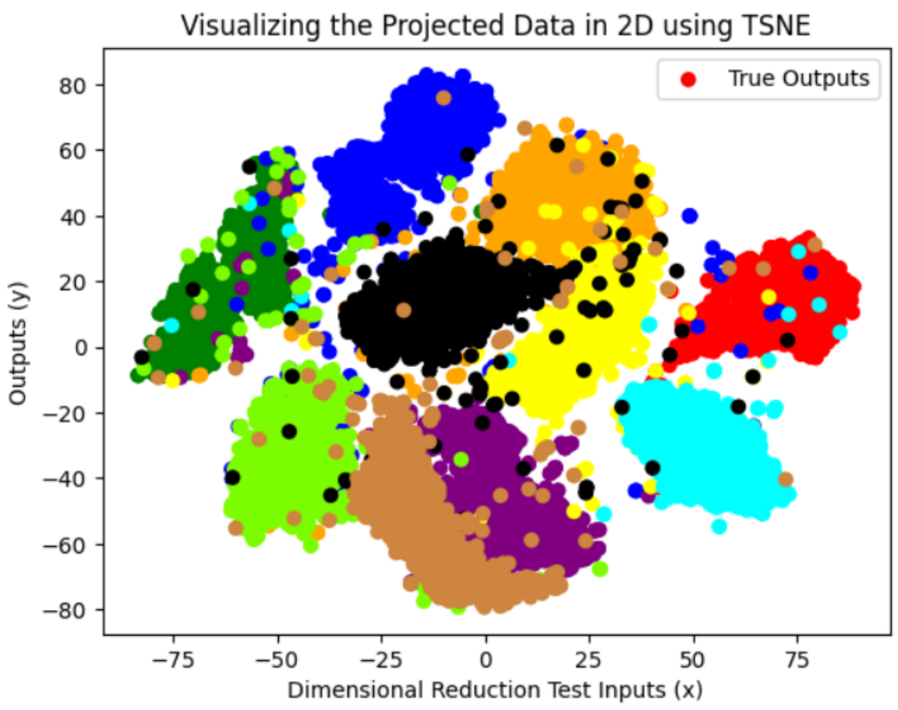


Figure 18: Visulaization the projected data in two dimensions using tSNE method