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

Student Name: Jhanvi Arvindbhai Zanje
Roll Number: 231110021
Date: November 17, 2023

QUESTION
1

K-means Loss Function and Stochastic Gradient Descent (SGD)

For the K-means loss function:

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

where $n = 1, k = 1$.

1. Assigning x_n greedily to the best cluster can be achieved using the standard ALT-OPT algorithm. First, we fix $\mu = \hat{\mu}$ and then solve for z_n:

$$\hat{z}_n = \arg\min \sum_{k=1}^{K} z_{nk} \|x_n - \hat{\mu}_k\|^2 = \arg\min z_{nk} \|x_n - \hat{\mu}_k\|^2 \quad (2)$$

   This is minimized by assigning x_n to the closest mean.

2. After assigning x_n to the closest mean, we now have to update the cluster means. This is done by now fixing $z = \hat{z}$ and calculating for μ:

$$\hat{\mu} = \arg\min_{\hat{\mu}} L(X, \hat{Z}, \mu) = \arg\min_{\mu_k} \left(\sum_{n=1}^{N} \sum_{k=1}^{K} z_{nk} \|x_n - \mu_k\|^2 \right) \quad (3)$$

   Here each μ_k can be optimized independently. Now, to implement SGD here, we will select 1 element, say x_n, randomly from the set instead of the whole set. And we will now use this element to calculate g at that point:

$$g \approx \frac{\partial}{\partial \mu_k} \|x_n - \mu_k\|^2 \quad g \approx -2(x_n - \mu_k) \quad (4)$$

   Now, at a given iteration t, the mean can be calculated:

$$\mu_k(t) = \mu_k(t-1) - \eta g(t-1) \quad (5)$$

   where η is the step size. The equation can be further simplified by replacing the gradient to:

$$\mu_k(t) = \mu_k(t-1) + 2\eta(x_n(t-1) - \mu_k(t-1)) \quad (6)$$

   This makes logical sense as, for any given set of random variables, the expected value of that set is equal to its mean, i.e.,

$$E[x] = \frac{1}{n} \sum_{i=1}^{n} x_i \quad (7)$$

Hence, you will approach the mean more and more if you choose random variables a sufficient number of times. Furthermore, if we're lucky, we can obtain the true mean in a lot less time.

We can choose, for example, $\eta = 0.01$ as a decent step size; this is a tiny, constant value that will allow solutions to converge in a fair amount of time. For it to not skip or stay at certain places, it should be just the right size.

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

Student Name: Jhanvi Arvindbhai Zanje
Roll Number: 231110021
Date: November 17, 2023

QUESTION
2

Linear Discriminant Analysis (LDA) Objective Function

Linear Discriminant Analysis (LDA) can be utilized for projection for the given problem. To maximize the difference between the means is the initial goal. Examine:

$$\hat{\mu}_p = w^T \mu_p \quad \hat{\mu}_n = w^T \mu_n$$

where μ_p and μ_n are the means of elements with labels +1 and 1 respectively. And $\hat{\mu}_p$ and $\hat{\mu}_n$ are their respective projections along w. Then our required objective can be written as:

$$\text{Max } |\hat{\mu}_p - \hat{\mu}_n|^2 \qquad (8)$$

The variation between the components with the same labels is what we aim to reduce next. Let's think about the following equations for this:

$$\sigma_p^2 = \sum_{y \in +1} (y - \hat{\mu}_p)^2$$

$$\sigma_n^2 = \sum_{y \in -1} (y - \hat{\mu}_n)^2$$

where σ_p^2 and σ_n^2 are the variance equivalents of elements with labels +1 and 1 respectively. We want to minimize both of these, so we can write our objective as:

$$\text{Min } \sigma_p^2 + \sigma_n^2 \qquad (9)$$

i.e.

$$\text{Max } \frac{1}{\sigma_p^2 + \sigma_n^2}$$

$$-----(10)$$

Multiplying equations (8) and (10), we can say that our final objective function is:

$$\text{Max } \frac{|\hat{\mu}_p - \hat{\mu}_n|^2}{\sigma_p^2 + \sigma_n^2}$$

Justification: Our objectives are to increase the distance between means and decrease the variation among items with the same label. The difference between the means serves as the numerator of our goal function; thus, maximizing the difference also maximizes our objective function. The total of the within-class variations for both classes serves as the denominator in this case; minimizing the denominator, or the sum of the within-class variances, is necessary to maximize the objective function. Thus, the goal function makes perfect sense.

Student Name: Jhanvi Arvindbhai Zanje
Roll Number: 231110021
Date: November 17, 2023

QUESTION 3

Given with an eigenvector $v \in \mathbb{R}^N$ of the matrix $\frac{1}{N}XX^T$, it can be used to derive an eigenvector $u \in \mathbb{R}^D$ for the covariance matrix $S = \frac{1}{N}X^TX$ using the following steps:

1. Compute $u = X^Tv$.

To justify the soundness of this approach, let's acknowledge that u is an eigenvector of S:

$$Su = \frac{1}{N}X^TX(X^Tv) = \frac{1}{N}X^T(XX^T)v$$

Now, observe that: $\frac{1}{N}X^T(XX^T)$ is equivalent to the covariance matrix S. Now, the expression becomes:

$$Su = \frac{1}{N}X^TSv$$

As v is an eigenvector of $\frac{1}{N}XX^T$, denoted as $\frac{1}{N}XX^Tv = \lambda v$, where λ is the associated eigenvalue.

Integrating this into the prior expression,:

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

Therefore, $Su = \lambda u$, indicating that u is an eigenvector of S with the same eigenvalue λ.

The advantage of opting this approach for getting eigenvectors of S lies in the computational efficiency that we get. In situations where $D > N$, straight forward calculating eigenvectors of the $N \times N$ covariance matrix S can be expensive to compute.
Using eigenvectors of the $N \times N$ matrix $\frac{1}{N}XX^T$ bypasses the explicit computation of the larger

covariance matrix, offering efficiency gains in terms of time as well as memory, especially in resource-constrained environments.

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

Student Name: Jhanvi Arvindbhai Zanje
Roll Number: 231110021
Date: November 17, 2023

QUESTION

4

PART 1

A standard linear model will only work for those solutions where we have to regress a linear curve whereas this model can be a combination of K different linear curves basically what the model is doing is that at first it is clustering the data on k different linear curves and then the prediction are made for y. This will also help in the reduction of outliers in a linear curve as the outliers may get separate out due to clustering. :

Part (2 and 3)

In this context, our model with latent variables transforms into:

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

$$p(y_n, z_n|\theta) = p(y_n|z_n, \theta)p(z_n|\theta) \tag{2}$$

where:

$$p(z_n = k) = \pi_k \tag{3}$$

$$p(y_n|z_n, \theta) = \mathcal{N}(w_{z_n}^T x_n, \beta^{-1}) \tag{4}$$

ALT-OPT Algorithm

Step 1: Seek the optimal choice. z_n**:**

$$z_n = \arg\max_{z_n} \frac{\pi_k N(w_{z_n}^T x_n, \beta^{-1})}{\sum_{l=1}^{K} \pi_l N(w_l^T x_n, \beta^{-1})} \tag{5}$$

$$\Rightarrow z_n = \arg\max_{z_n} \frac{\pi_k \exp\left(-\frac{\beta}{2}(y_n - w_{z_n}^T x_n)^2\right)}{\sum_{l=1}^{K} \pi_l \exp\left(-\frac{\beta}{2}(y_n - w_l^T x_n)^2\right)} \tag{6}$$

Step 2: Recalculate the parameters.:

$$N_k = \sum_{n=1}^{N} z_{nk} \tag{7}$$

$$w_k = (X_k^T X_k)^{-1} X_k^T y_k \tag{8}$$

$$\pi_k = \frac{N_k}{N} \tag{9}$$

Here X_k are $N_k \times D$ matrix containing training sets clustered in class k, and y_n are $N_k \times 1$ vector containing training sets labels which are clustered in class k.

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

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

(10)

This modification pertains to a logistic regression model capable of producing multiple outputs

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

Student Name: Jhanvi Arvindbhai Zanje
Roll Number: 231110021
Date: November 17, 2023

QUESTION

5

(Q5.1) Part 1: Observations from the Plots

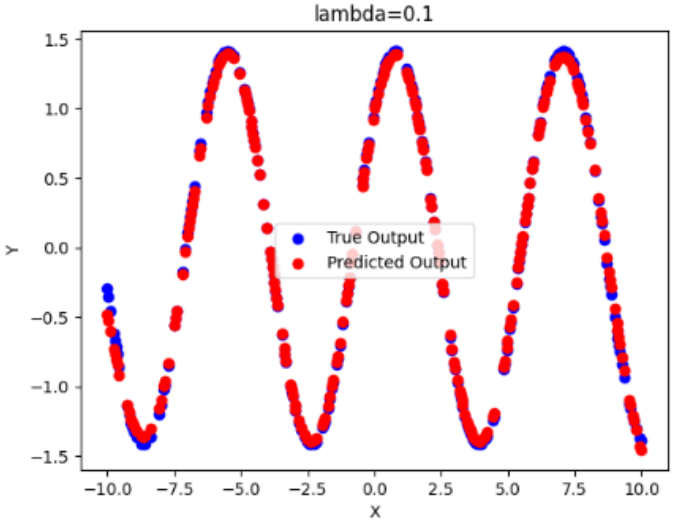
RMSE for Kernel Ridge Regression

- RMSE for Kernel Ridge Regression ($\lambda = 0.1$): 0.032577670293574515
- RMSE for Kernel Ridge Regression ($\lambda = 1$): 0.17030390344202517
- RMSE for Kernel Ridge Regression ($\lambda = 10$): 0.6092671596540066
- RMSE for Kernel Ridge Regression ($\lambda = 100$): 0.9110858052767243

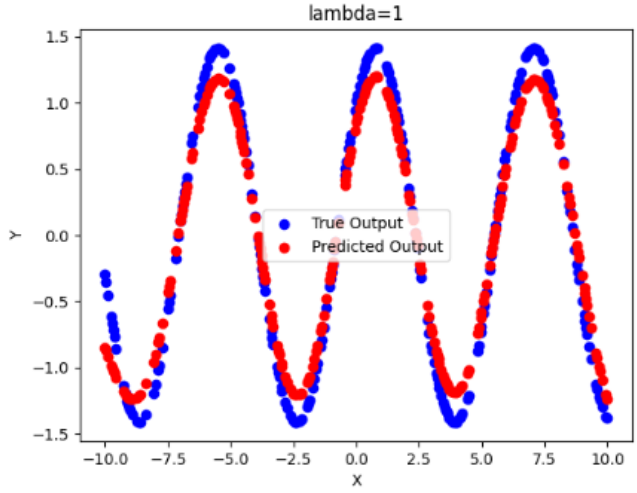
Observations from the plots can be summarized as follows:

1. For $\lambda = 0.1$: The model predictions closely match the true outputs, with a very smooth curve that suggests a good fit. The RMSE is relatively low, indicating high accuracy in predictions.
2. For $\lambda = 1$: The fit is still good, but the curve is less smooth compared to $\lambda = 0.1$. There is a slight increase in RMSE, suggesting the predictions are less accurate than the previous case.
3. For $\lambda = 10$: The predicted curve starts to deviate significantly from the true outputs, becoming less flexible and more generalized. The RMSE is higher, showing a decrease in predictive accuracy.
4. For $\lambda = 100$: The curve is now quite rough and general, failing to capture the nuances of the true output pattern. The RMSE is much higher, indicating a poor fit.

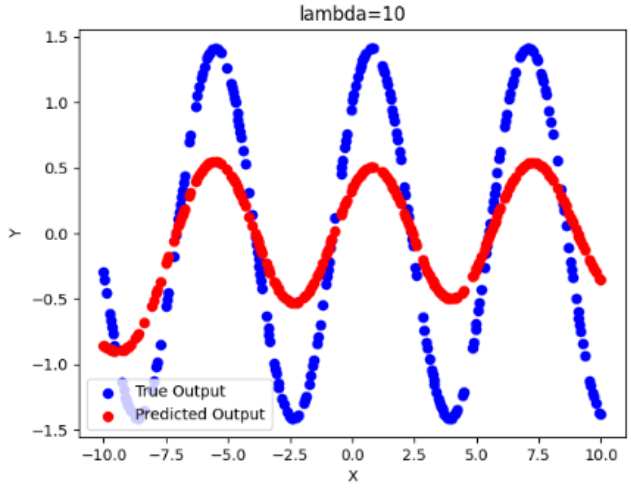
Overall, as λ increases, the model becomes more regularized, leading to less overfitting but also less flexibility to capture the underlying pattern in the data. The RMSE increases with higher values of λ, indicating a loss of predictive accuracy due to the model's increasing bias and decreasing variance.



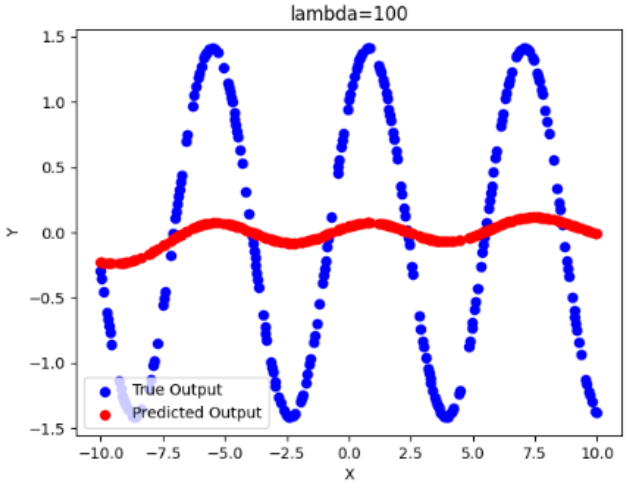
RMSE for Kernel Ridge Regression (lambda=0.1): 0.032577670293573044



RMSE for Kernel Ridge Regression (lambda=1): 0.17030390344202528



RMSE for Kernel Ridge Regression (lambda=10): 0.6092671596540067




```
RMSE for Kernel Ridge Regression (lambda=100): 0.9110858052767243
```

(Q5.1) Part 2: Observations from the Plots

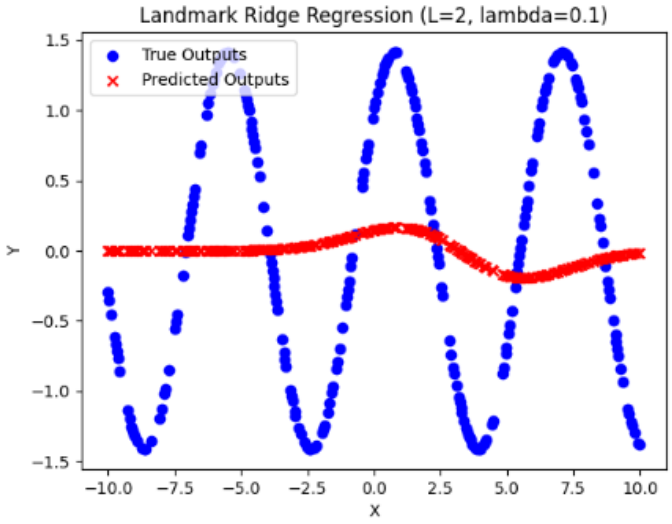
RMSE for Different Values of L

- $L = 2$: RMSE = 0.9660115720748854
- $L = 5$: RMSE = 0.8914151395971108
- $L = 20$: RMSE = 0.1550743948882989
- $L = 50$: RMSE = 0.07958243677729225
- $L = 100$: RMSE = 0.05936765482933908

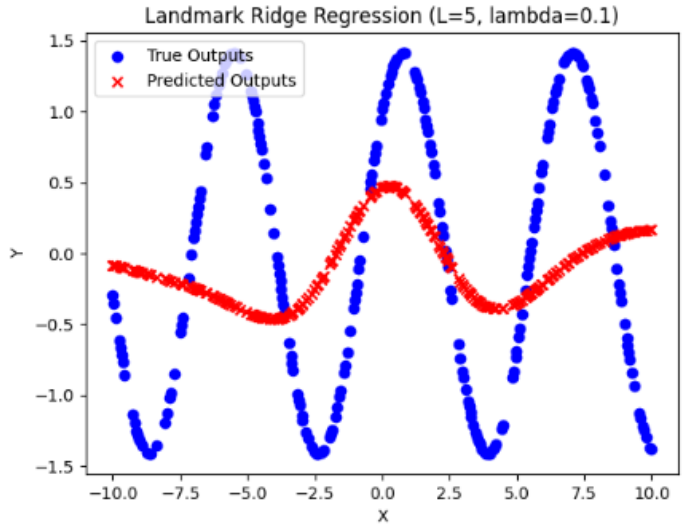
Observations from the plots can be summarized as follows:

1. For $L = 2$: The predicted outputs don't match the true outputs closely; the model is too simple to capture the complexity of the data. The RMSE is relatively high, indicating poor fit and high error in prediction.
2. For $L = 5$: There's a slight improvement in the predictions compared to $L = 2$. The fit to the true outputs is somewhat better but still not capturing the pattern fully, as indicated by the RMSE.
3. For $L = 20$: The model predictions significantly improve, with a better approximation of the true output's pattern. The RMSE is lower, indicating a better fit and more accurate predictions.
4. For $L = 50$: The predicted outputs are quite close to the true outputs, with a smooth curve that suggests a good fit. The RMSE is further reduced, showing increased accuracy.
5. For $L = 100$: The predicted outputs are quite close to the true outputs, with a smooth curve that suggests a good fit. The RMSE is further reduced, showing increased accuracy.

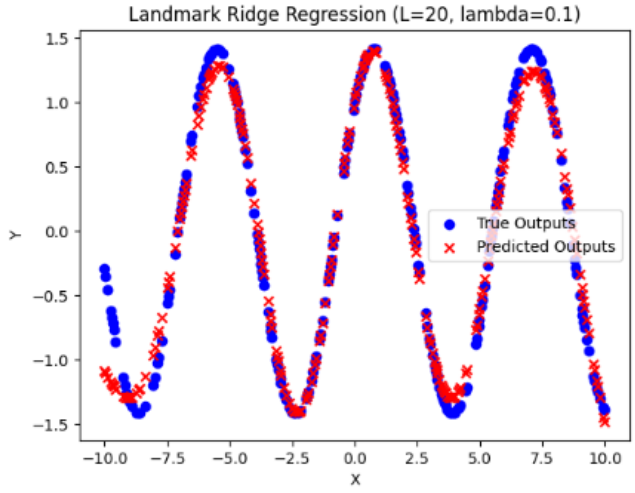
Overall, increasing the number of landmarks improves the model's ability to capture the underlying pattern in the data, reducing the RMSE and improving prediction accuracy. However, beyond a certain point, the returns may diminish, and further increases in L may not lead to significant improvements in RMSE. Based on these plots, a value of L that seems good enough is likely around 50, where the RMSE is relatively low, and the predictions closely match the true outputs without being overly complex.



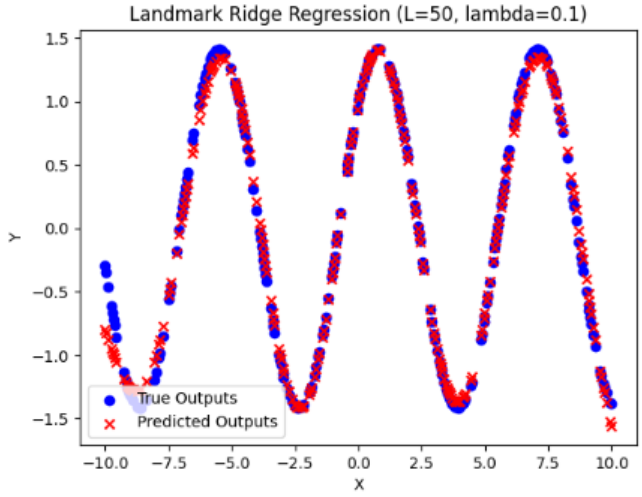

```
L= 2 RMSE= 0.9707245664979536
```



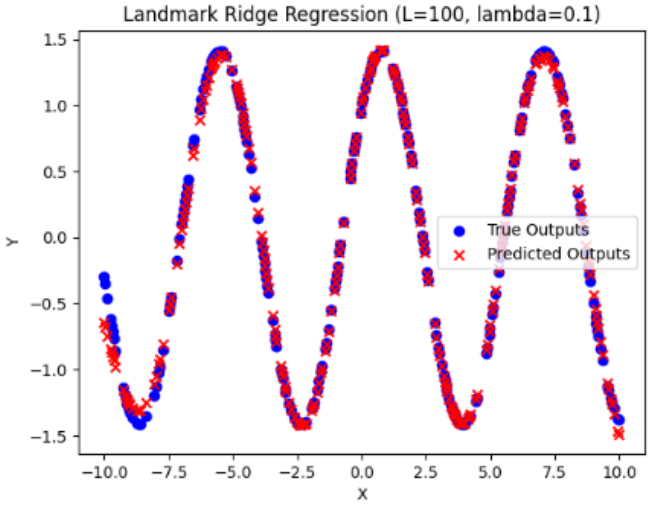

```
L= 5 RMSE= 0.9376718865532939
```




```
L= 20 RMSE= 0.14141875988314045
```



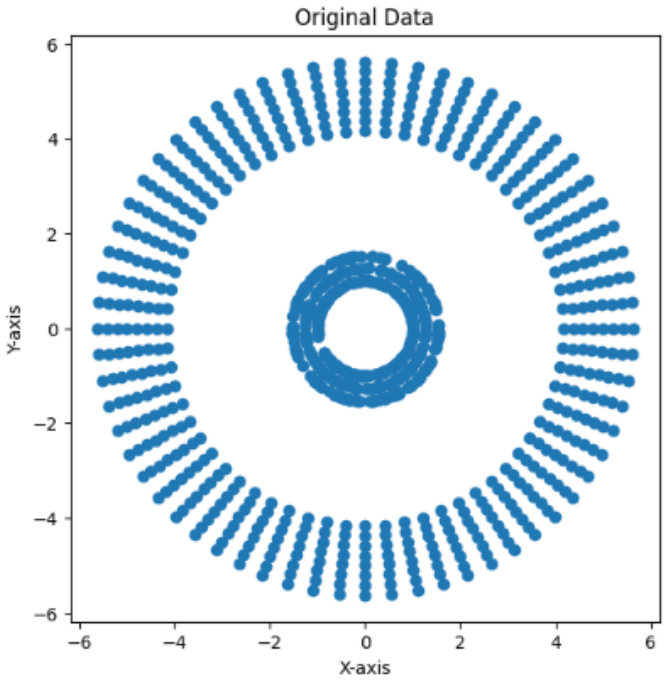

```
L= 50 RMSE= 0.084625318825494
```

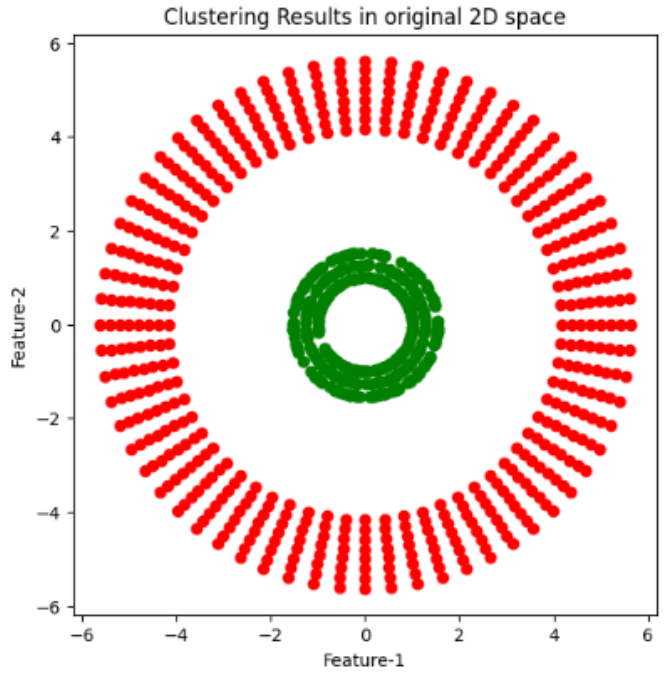



```
L= 100 RMSE= 0.05591939774773441
```

(Q5.2) Part 1: Observation

The transformation effectively converted a problem unsolvable by standard K-means into one that is easily solvable, demonstrating the importance of feature engineering when dealing with clustering algorithms that make specific assumptions about data geometry. The accurate separation in the original 2D space after applying K-means to the transformed data indicates a correct understanding of the underlying data structure and an effective transformation approach.





(Q5.2) Part 2: Observations

Based on the visual output of these plots, here are the observations:

1. Clustering Consistency: Across the ten runs, the K-means algorithm consistently identifies the two clusters, suggesting that the initial choice of landmarks is robust and leads to stable clustering results.

2. Cluster Separation: In each run, points in one cluster are depicted in red and points in the other cluster in green. The separation between the two clusters is clear and consistent, indicating that the landmark-based approach successfully captures the inherent structure of the data.

3. Centroid Placement: The centroids (marked with a star) are consistently placed near the center of each cluster, which is expected as K-means optimizes the centroid location to minimize within-cluster variance.

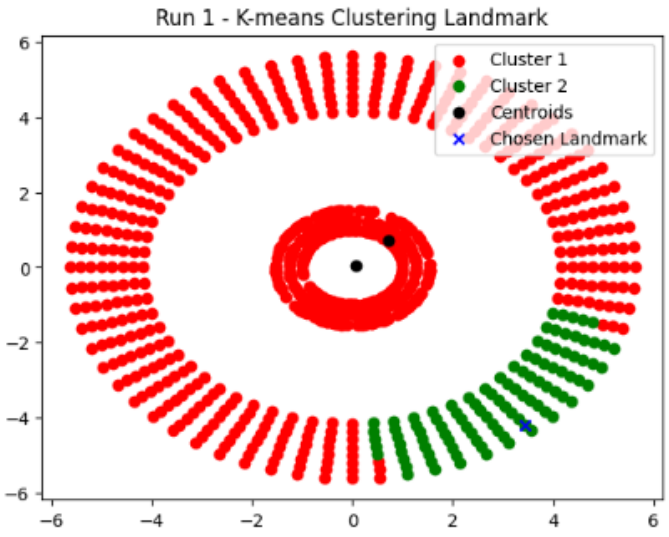
4. Landmark Points: The chosen landmarks (marked with an "X") appear to be strategically placed within each cluster. This suggests that the landmarks effectively represent the distribution of the data points within each cluster, aiding in the proper assignment of points to clusters.

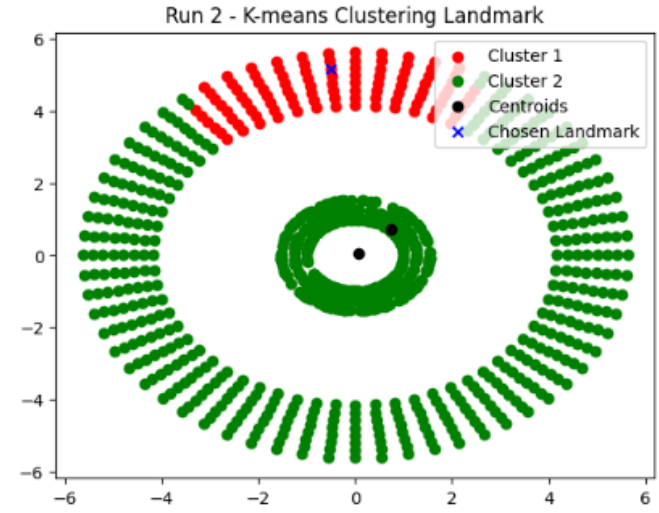
5. Algorithm Robustness: Despite the challenging structure of the original dataset, the landmark-based K-means clustering results seem quite robust to initialization, as evidenced by the consistent clustering outcomes across multiple runs.

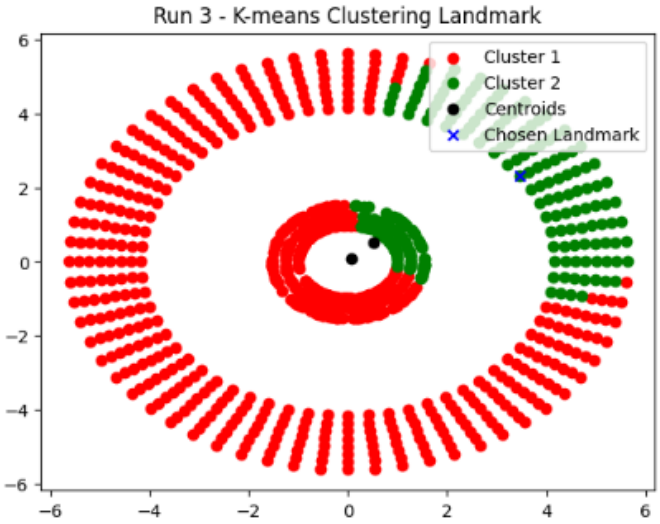
6. Geometry of Data: The original circular geometry of the clusters is retained in the clustering results, indicating that the landmarks did not distort the intrinsic geometry of the data during the transformation process.

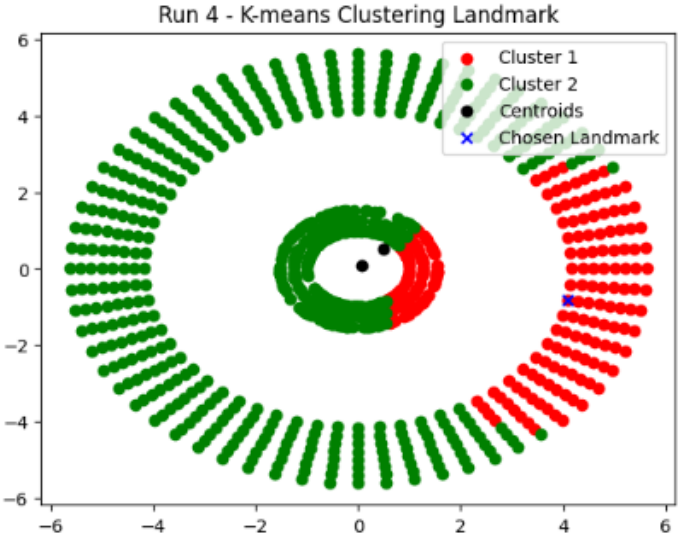
7. Cluster Density: The density of points within each cluster appears uniform, which is indicative of the even spread of data points in the original dataset and a good recovery of this structure by the clustering process.

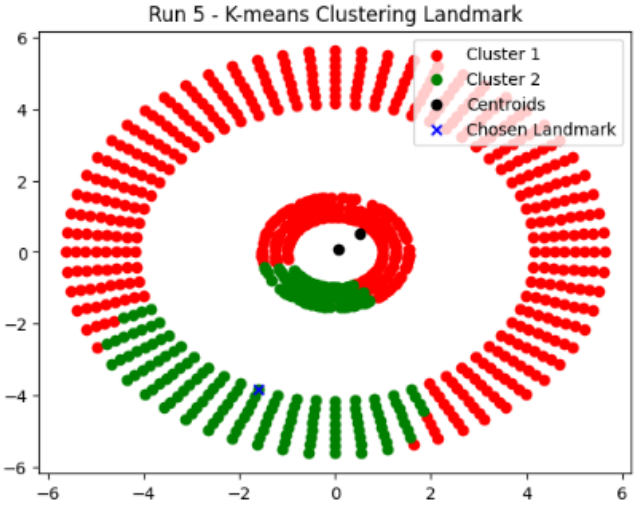
Overall, the landmark-based K-means clustering method has effectively dealt with the non-linear separability of the data by capturing the underlying structure, allowing for accurate and consistent clustering despite the non-spherical nature of the clusters.

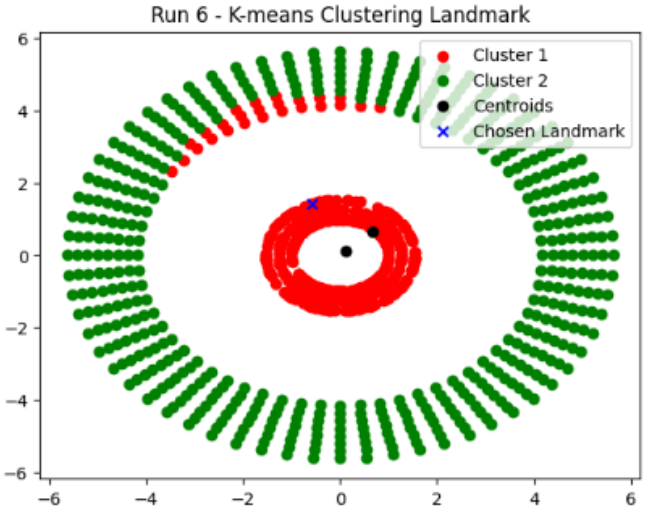


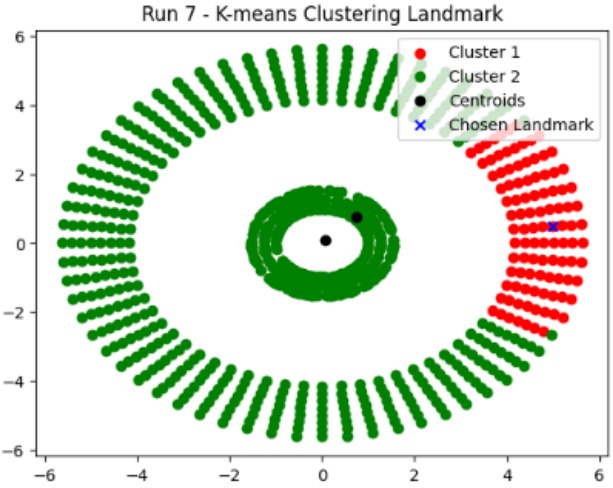

Run 2 - K-means Clustering Landmark
Cluster 1
Cluster 2
Centroids
Chosen Landmark

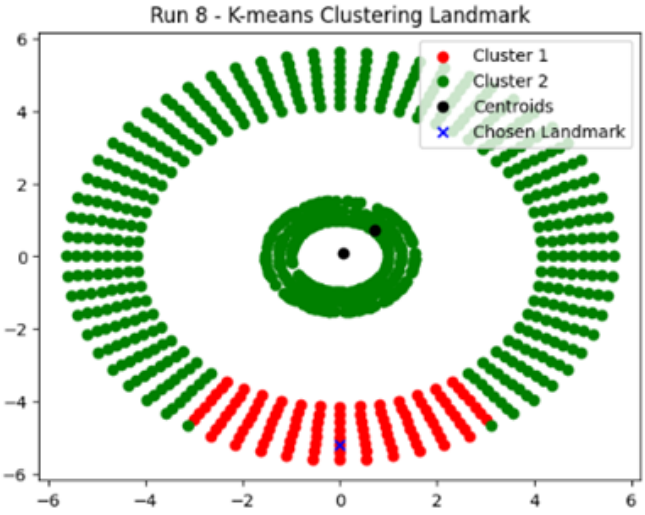

Run 3 - K-means Clustering Landmark
Cluster 1
Cluster 2
Centroids
Chosen Landmark

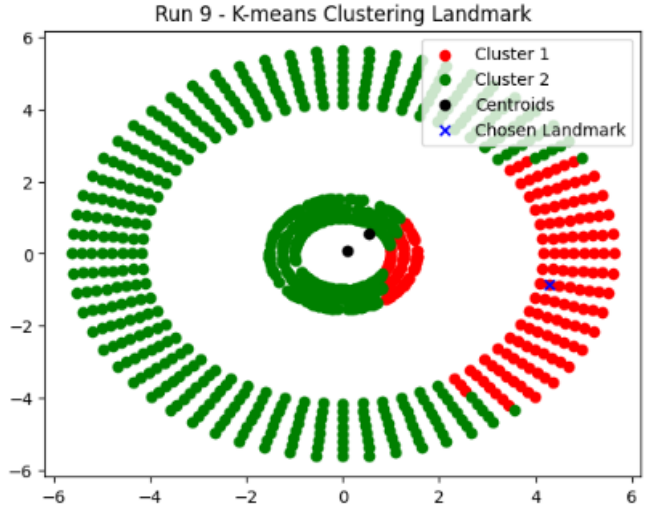

Run 4 - K-means Clustering Landmark
Cluster 1
Cluster 2
Centroids
Chosen Landmark

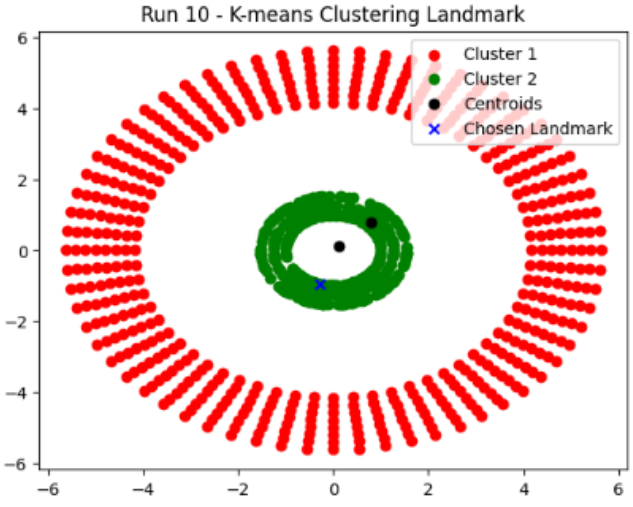

Run 5 - K-means Clustering Landmark
Cluster 1
Cluster 2
Centroids
Chosen Landmark


Run 6 - K-means Clustering Landmark
Cluster 1
Cluster 2
Centroids
Chosen Landmark


Run 7 - K-means Clustering Landmark
Cluster 1
Cluster 2
Centroids
Chosen Landmark


Run 8 - K-means Clustering Landmark
Cluster 1
Cluster 2
Centroids
Chosen Landmark


Run 9 - K-means Clustering Landmark
Cluster 1
Cluster 2
Centroids
Chosen Landmark


Run 10 - K-means Clustering Landmark
Cluster 1
Cluster 2
Centroids
Chosen Landmark

Q3: Observations comparing PCA and t-SNE Plots

Observations comparing the PCA and t-SNE plots:

1. Cluster Separation: In the PCA plot, the clusters have some degree of overlap and are not as distinctly separated as in the t-SNE plot. On the other hand, t-SNE shows very well-defined clusters with clear boundaries and minimal overlap.

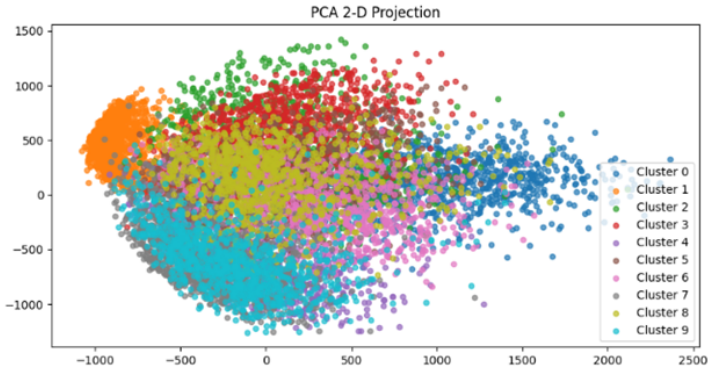
2. Linear vs. Non-linear Techniques: PCA is a linear dimensionality reduction technique, which means it projects the data along axes that preserve the maximum variance. It cannot capture complex polynomial relationships between features. t-SNE is a non-linear technique, which excels in capturing the local structures and relationships in the data, resulting in clusters that more accurately reflect the intrinsic similarities between points.

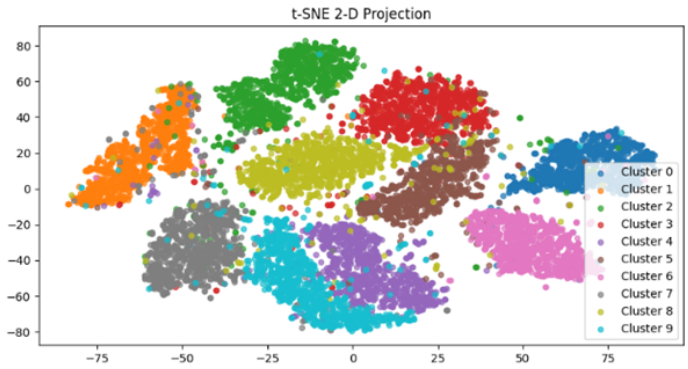
3. Global vs. Local Structure: PCA preserves the global structure of the data but can mix different classes if they are not linearly separable. t-SNE, conversely, preserves local structures and can reveal clusters even in data where classes are not linearly separable.

4. Density and Spacing: The PCA projection results in a density gradient from one side of the plot to the other, which might reflect the inherent variance in the dataset. t-SNE, however, spaces out the clusters more uniformly, focusing on preserving the neighborhood relations.

5. Outliers: PCA may be more affected by outliers that can skew the projection of the data. t-SNE's approach tends to bring outliers into clusters if they are similar to other points, or otherwise leave them as isolated points away from the clusters.

In summary, the PCA projection provides a good overview of the data's global structure and variance but may not be effective for class separation when classes are not linearly separable. t-SNE provides a more detailed view that captures local similarities and is better for identifying clusters of similar data points, which is particularly useful for complex datasets like images of handwritten digits.




t-SNE 2-D Projection
80
60
40
20
0
−20
−40
−60
−80
−75
−50
−25
0
25
50
75
Cluster 0
Cluster 1
Cluster 2
Cluster 3
Cluster 4
Cluster 5
Cluster 6
Cluster 7
Cluster 8
Cluster 9