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

Student Name: Ritick Gupta
Roll Number: 200801
Date: November 17, 2023

QUESTION 1

In the online variant of the K-means algorithm employing Stochastic Gradient Descent (SGD), the updating of cluster assignments and means occurs individually for each data point. Let's delve into each stage:

1: Computing Distances

- Calculate the Euclidean distance between the current data point x_n and each of the cluster centers μ_k for $k = 1, 2, \ldots, K$.
$$d_{nk} = \|x_n - \mu_k\|^2$$

2: Assigning Data Points to Clusters (SGD-based Assignment)

- In the initial step, assign each data point x_n to the cluster with the closest mean using the following rule:
$$z_{nk} = \begin{cases} 1 & \text{if } k = \text{argmin}_j d_{nj} \\ 0 & \text{otherwise} \end{cases}$$
This ensures associating each point with the cluster whose mean minimizes the squared Euclidean distance.

3: SGD-based Cluster Mean Update Equations

- Define the objective function L as:
$$L = \sum_{n=1}^{N} \sum_{k=1}^{K} z_{nk} \cdot \|x_n - \mu_k\|^2$$
- Update the cluster means using SGD by computing the gradient of L with respect to μ_k:
$$\nabla_{\mu_k} L = -2 \sum_{n=1}^{N} z_{nk} \cdot (x_n - \mu_k)$$
- The update equation for μ_k is:
$$\mu_k^{(\text{t+1})} = \mu_k^{(\text{t})} + 2\eta \sum_{n=1}^{N} z_{nk} \cdot (x_n - \mu_k^{(\text{t})})$$
Here, t denotes the iteration number, η is the learning rate, and the update is applied for each data point x_n as it is processed.

Intuition for the Update Equation

- The update equation intuitively propels the cluster mean μ_k towards minimizing the distance between the current data point x_n and the current cluster mean.
- The negative gradient term directs movement in the direction of decreasing L, aligning with the goal of minimizing the K-means objective function.

Choice for appropriate Step Size (η)

- Choosing an appropriate step size is pivotal for SGD convergence.
  As the algorithm progresses, the steps become smaller, allowing the algorithm to converge more precisely to a minimum.
- A commonly used strategy is to use a diminishing step size, where the step size decreases over time. One common form for a diminishing step size is:

  $$\eta_t = \frac{\eta_0}{t}$$

  where:

  - η_t is the learning rate at iteration t,
  - η_0 is the initial learning rate,
  - t is the iteration number or the number of data points processed.
- A diminishing step size can enhance the stability of the optimization process. In the initial phases, when the algorithm is distant from convergence, larger steps may accelerate the progress. As it approaches a minimum, reducing the step size becomes crucial to prevent overshooting and ensure a more controlled convergence.
- However, the optimal step size may vary based on data and problem characteristics, necessitating experimentation to find the most effective value.

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

Student Name: Ritick Gupta
Roll Number: 200801
Date: November 17, 2023

QUESTION
2

Objective Function for Fisher Discriminant Analysis

Suppose we are given some labeled training data $\{(x_n, y_n)\}$ with inputs $x_n \in \mathbb{R}^D$ and labels $y_n \in \{-1, +1\}$.

Let:

- w be the projection vector in $\mathbb{R}^D$.
- x_n be the input in $\mathbb{R}^D$.
- y_n be the label for the input ($y_n \in \{-1, +1\}$).

We aim to project the inputs onto a one-dimensional space using a projection vector $w \in \mathbb{R}^D$. This projection is designed so that, upon completion, the distance between the means of inputs from the two classes is maximized, while simultaneously minimizing the distances between inputs within each class.

The objective function that achieves this goal is often formulated using the Fisher Discriminant Analysis (FDA) or Linear Discriminant Analysis (LDA) framework.
It is typically formulated as the ratio of the between-class variance to the within-class variance. Mathematically, the objective function for Fisher's criterion is:

$$J(w) = \frac{w^T S_B w}{w^T S_W w}$$

Where:

- S_B is the between-class scatter matrix, defined as $\sum_{c=1}^{C} N_c(\mu_c - \mu)(\mu_c - \mu)^T$, where C is the number of classes, N_c is the number of samples in class c, μ_c is the mean of class c, and μ is the overall mean of the data.
- S_W is the within-class scatter matrix, defined as $\sum_{c=1}^{C} \sum_{i=1}^{N_c} (x_i - \mu_c)(x_i - \mu_c)^T$, where x_i is a sample from class c, μ_c is the mean of class c.

Maximizing $J(w)$ will ensure that the means of the projected data from different classes are far apart (maximizing the between-class variance), while also minimizing the variance within each class (minimizing the within-class variance).
This objective helps in finding a discriminative direction for projection.

Optimizing this objective function leads to finding the optimal projection vector w that achieves the desired goal of maximizing the separation between classes while minimizing the scatter within classes.

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

Student Name: Ritick Gupta
Roll Number: 200801
Date: November 17, 2023

QUESTION 3

Consider a centered data matrix X of dimensions $N \times D$ (where $D > N$), and let S denote its covariance matrix, given by $S = \frac{1}{N} X^T X$.
Suppose we are provided with an eigenvector $v \in \mathbb{R}^N$ corresponding to the matrix $S' = \frac{1}{N} X X^T$.

The objective is to establish that by utilizing v, we can derive an eigenvector $u \in \mathbb{R}^D$ for S.

Commencing with the eigenvalue equation for S':

$$S'v = \lambda v$$

Here, v is the eigenvector, and λ is the associated eigenvalue. Expressing $S'v$ using the definition of S':

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

Now, pre-multiply both sides by X^T:

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

Observing that $X^T v$ is a D-dimensional vector, this expression closely mirrors the eigenvalue equation for the matrix $\frac{1}{N} X^T X$. Given u as an eigenvector of $\frac{1}{N} X^T X$ (i.e., $\frac{1}{N} X^T X u = \lambda u$), we can express $X^T v$ in terms of u:

$$u = X^T v$$

The advantage of this approach lies in the computational efficiency gained by directly working with the smaller covariance matrix S' instead of the larger matrix $X^T X$. This simplification proves particularly advantageous when D significantly exceeds N, reducing the computational cost associated with principal component analysis (PCA).
The complexity to compute $S = \frac{1}{N} X^T X$ is $O(KD^2)$ whereas for $S' = \frac{1}{N} X X^T$ is $O(KN^2) + O(KND) = O(KND)$ for $D > N$ which is less than $O(KD^2)$.

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

Student Name: Ritick Gupta
Roll Number: 200801
Date: November 17, 2023

QUESTION 4

The model introduces latent variables z_n, which represent the cluster assignment for each data point (x_n, y_n). It assumes a multinomial prior $p(z_n)$ with parameters $\pi_1, \ldots, \pi_K$. Each cluster is linked to a distinct weight vector w_k from the set $W = [w_1, w_2, \ldots, w_K]$.

The response y_n, given the cluster assignment z_n, is modeled using a probabilistic linear model. This flexibility allows the model to capture K distinct patterns for different clusters, providing a more versatile representation compared to a standard probabilistic linear model.

This adaptability enables the model to adjust to various subgroups within the data, potentially enhancing overall performance.

The ALT-OPT algorithm alternates between updating latent variables Z and global parameters $\Theta = \{(w_1, \ldots, w_K), (\pi_1, \ldots, \pi_K)\}$. The update equations for each step are as follows:

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

where:

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

ALT-OPT Algorithm:
Step 1: Finding the best z_n

$$\begin{aligned}\hat{z}_n &= \arg\max_{z_n} p(z_n|y_n, \theta) \\ &= \arg\max_{z_n} \frac{p(z_n = k)p(y_n|z_n = k, \theta)}{\sum_{l=1}^{K} p(z_n = l)p(y_n|z_n = l, \theta)} \\ &= \arg\max_{z_n} \frac{\pi_k N(w_{z_n}^T x_n, \beta^{-1})}{\sum_{l=1}^{K} \pi_l N(w_l^T x_n, \beta^{-1})}\end{aligned}$$

Step 2: Re-estimating the parameters:

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

$$w_k = (X_k^T X_k)^{-1} X_k^T y_k$$
$$\pi_k = N_k/N$$

where X_k is a $N_k \times D$ matrix containing training sets clustered in class k, and y_k is a $N_k \times 1$ vector containing training set labels clustered in class k.

If $\pi_k = 1/K$ then:

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

Intuition: ALT-OPT functions analogously to the EM algorithm. During the E-step, the algorithm probabilistically assigns each data point to clusters based on the current model, while in the M-step, it updates the parameters relying on the expected values derived from the latent variables.

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

Student Name: Ritick Gupta
Roll Number: 200801
Date: November 17, 2023

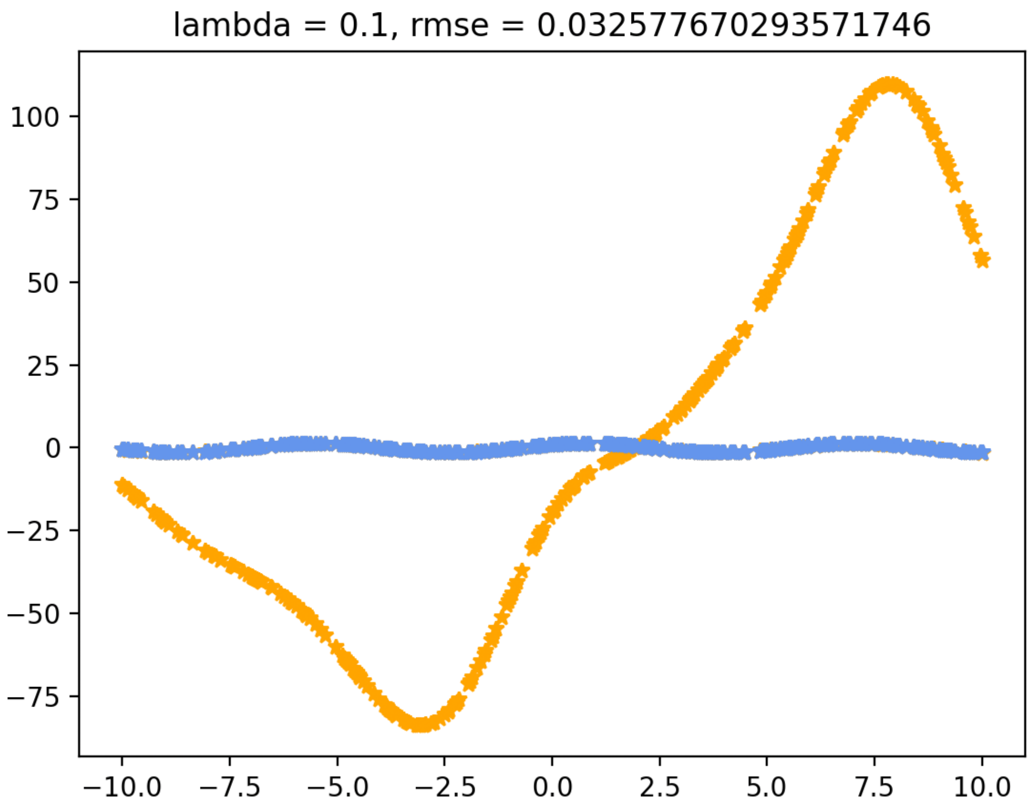
QUESTION 5

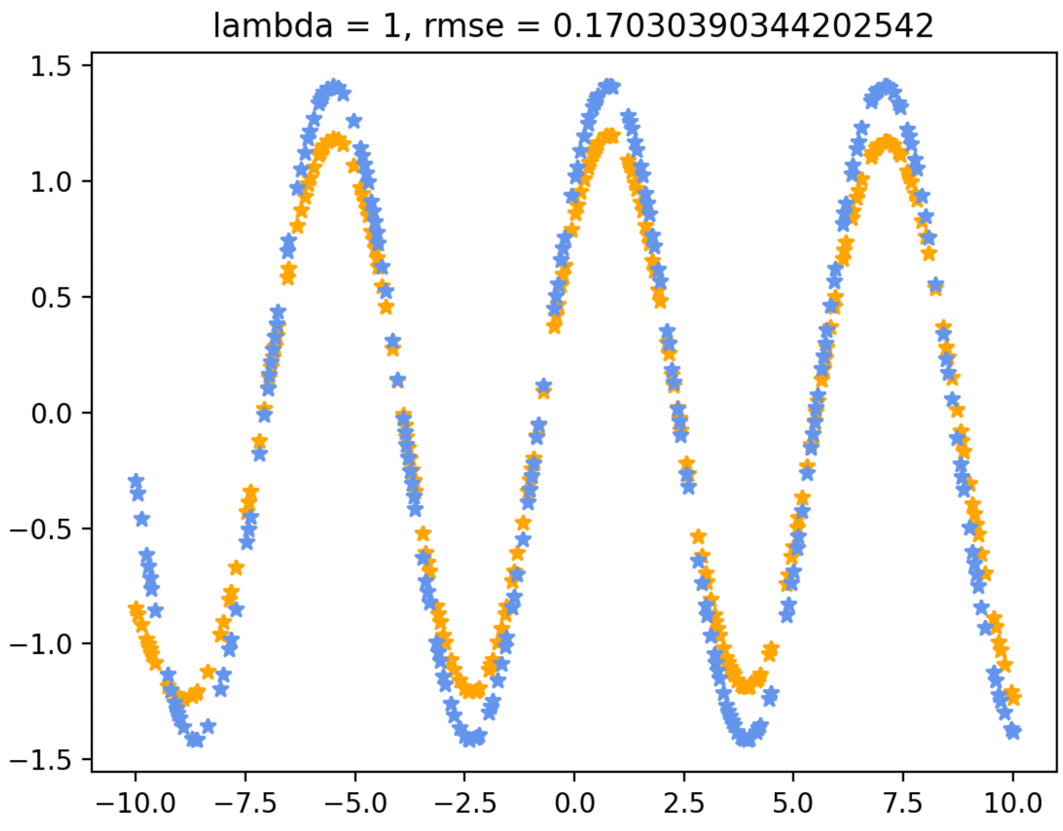
My solution to problem 5

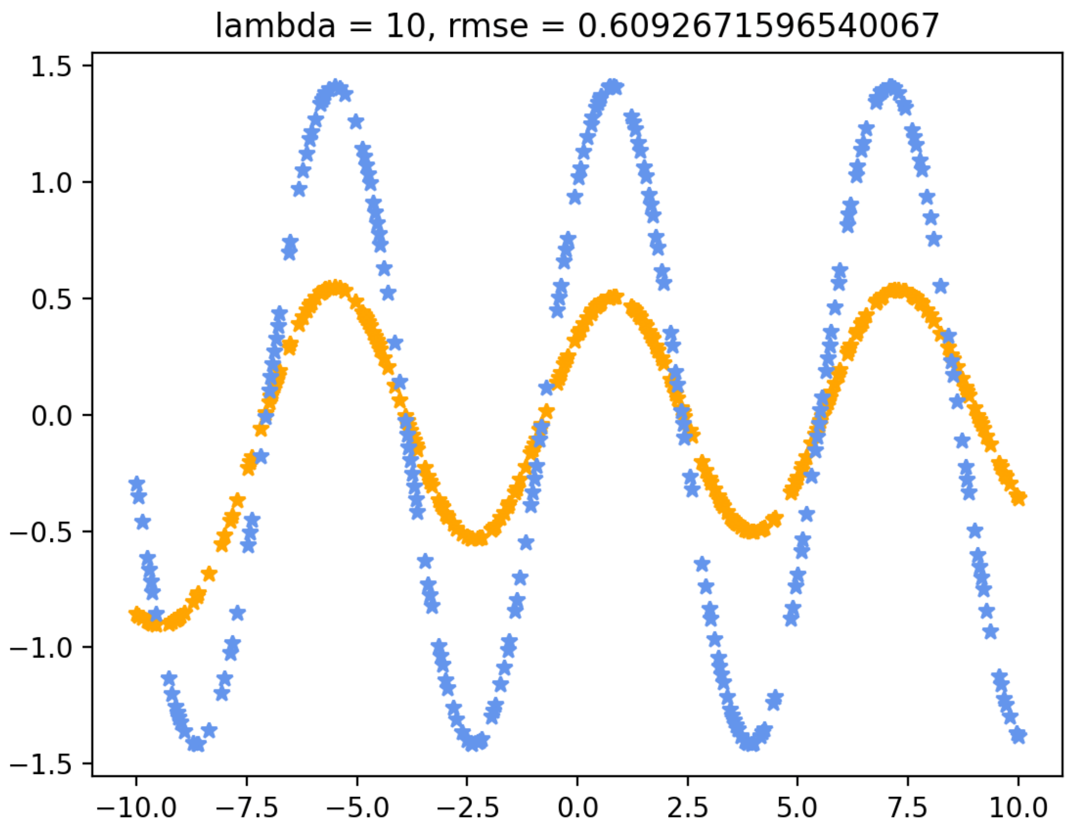
The plots for all the parts of the questions are give below

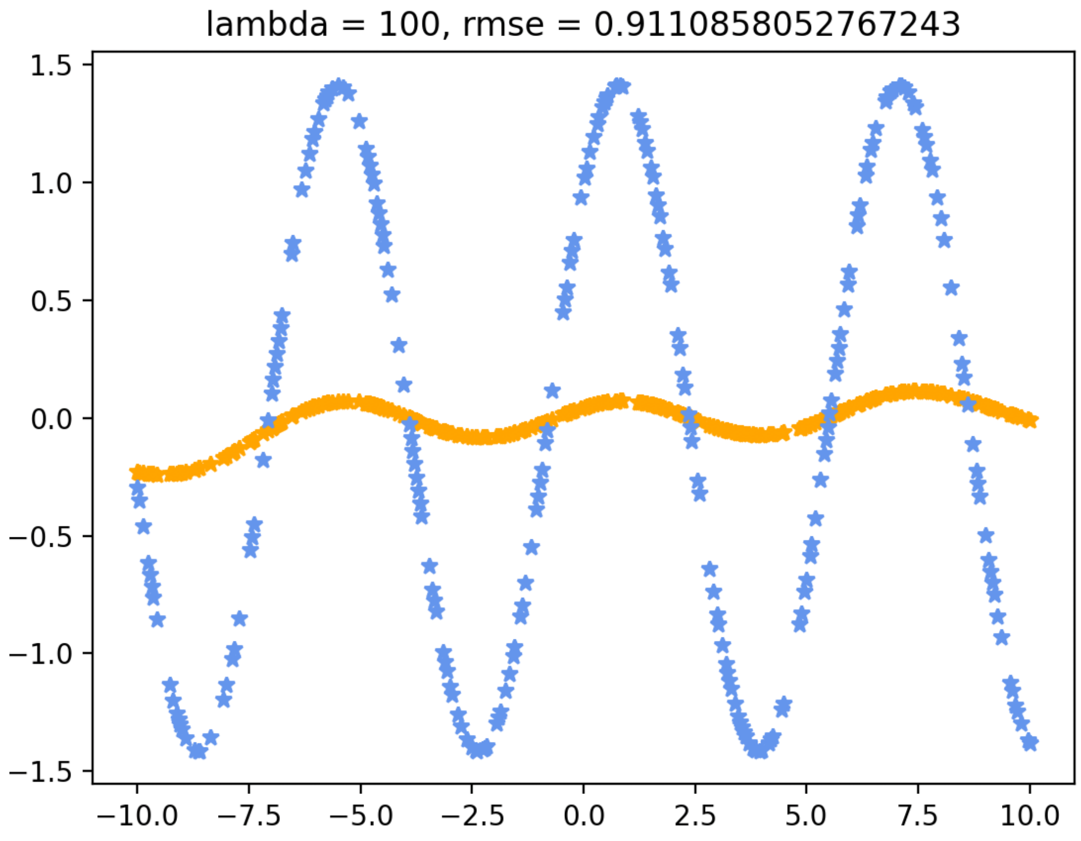
Part 1:

i) Kernel Ridge regression

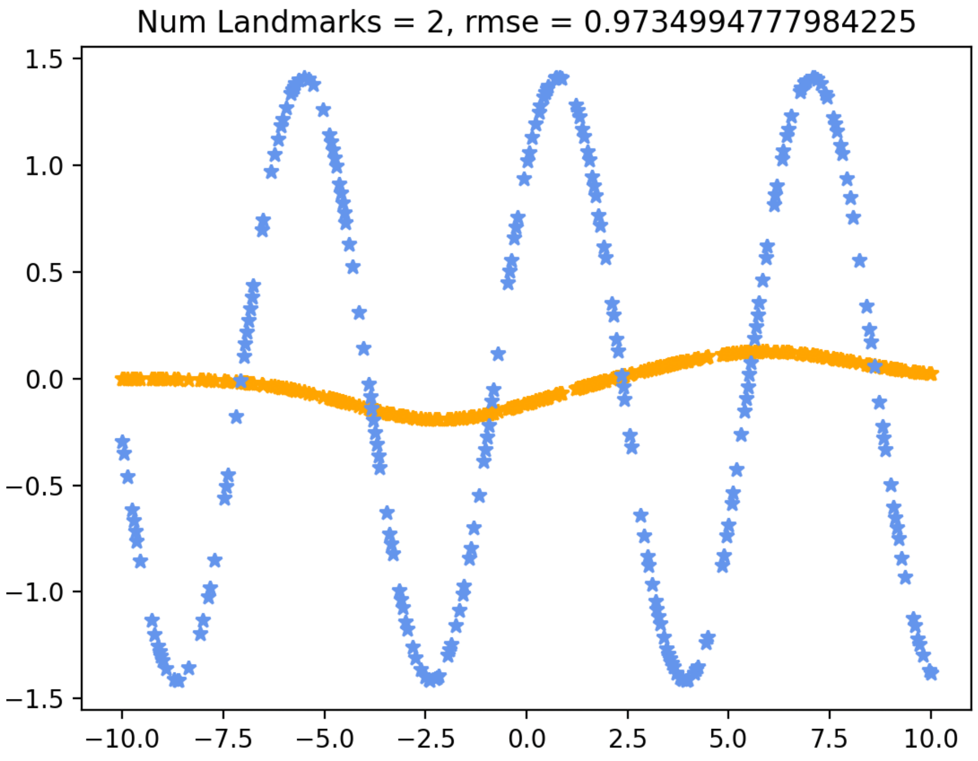


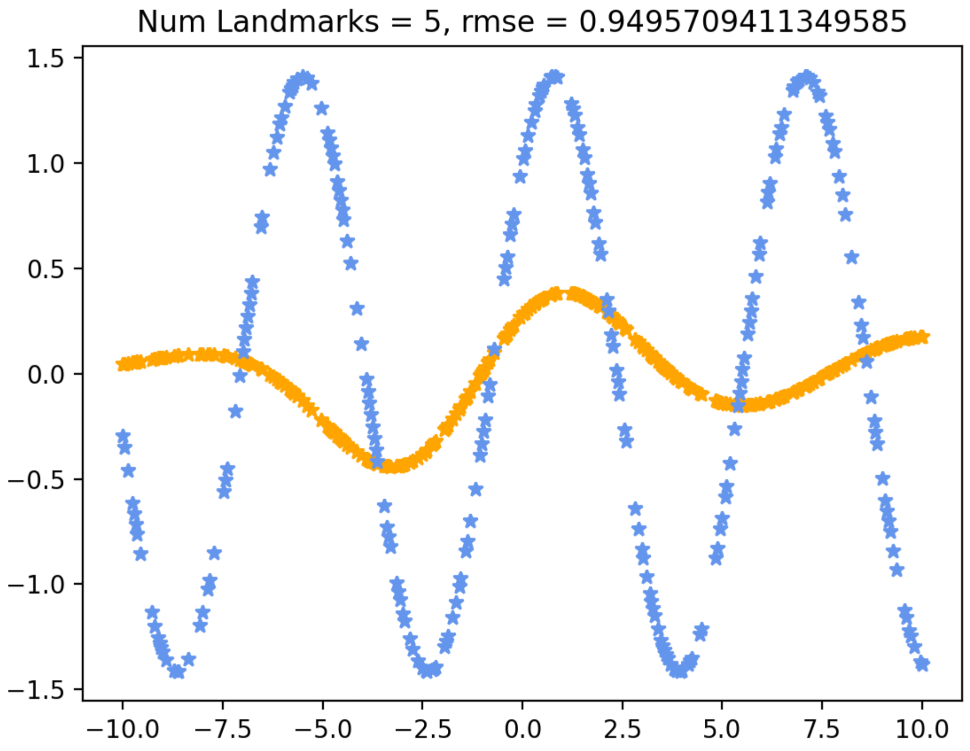


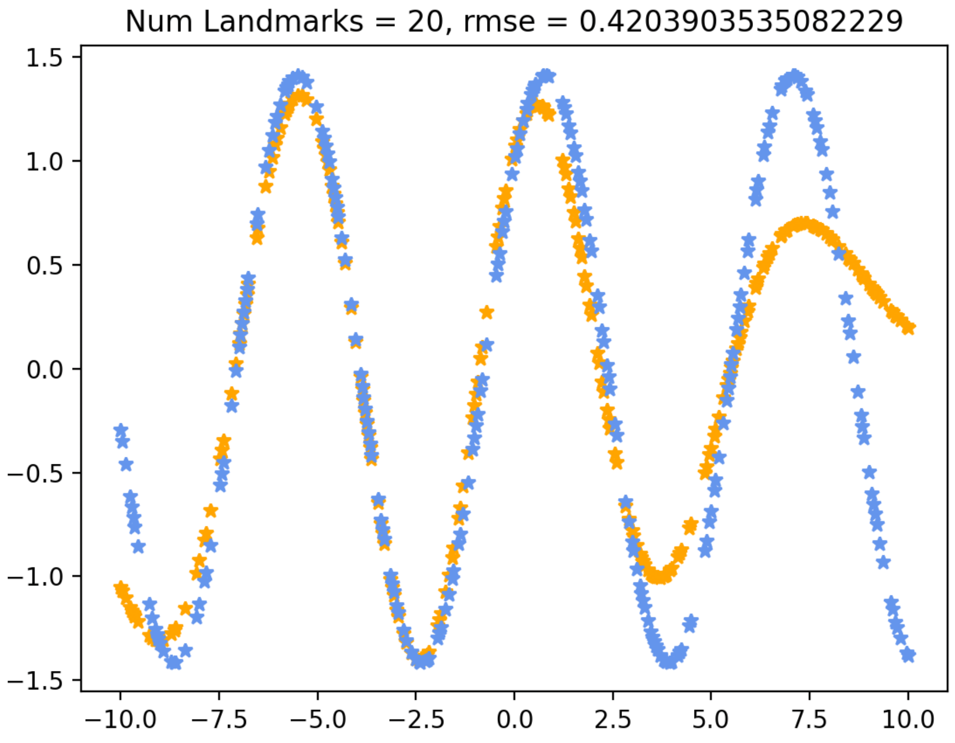

lambda = 10, rmse = 0.6092671596540067
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−10.0
−7.5
−5.0
−2.5
0.0
2.5
5.0
7.5
10.0

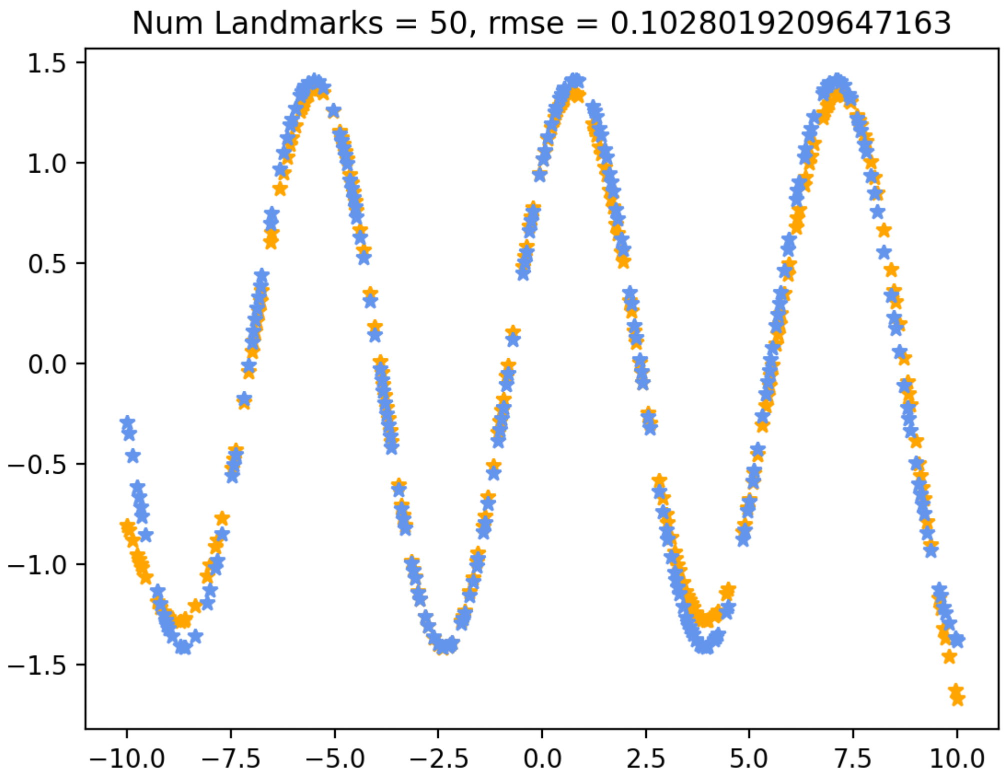

lambda = 100, rmse = 0.9110858052767243
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−10.0
−7.5
−5.0
−2.5
0.0
2.5
5.0
7.5
10.0

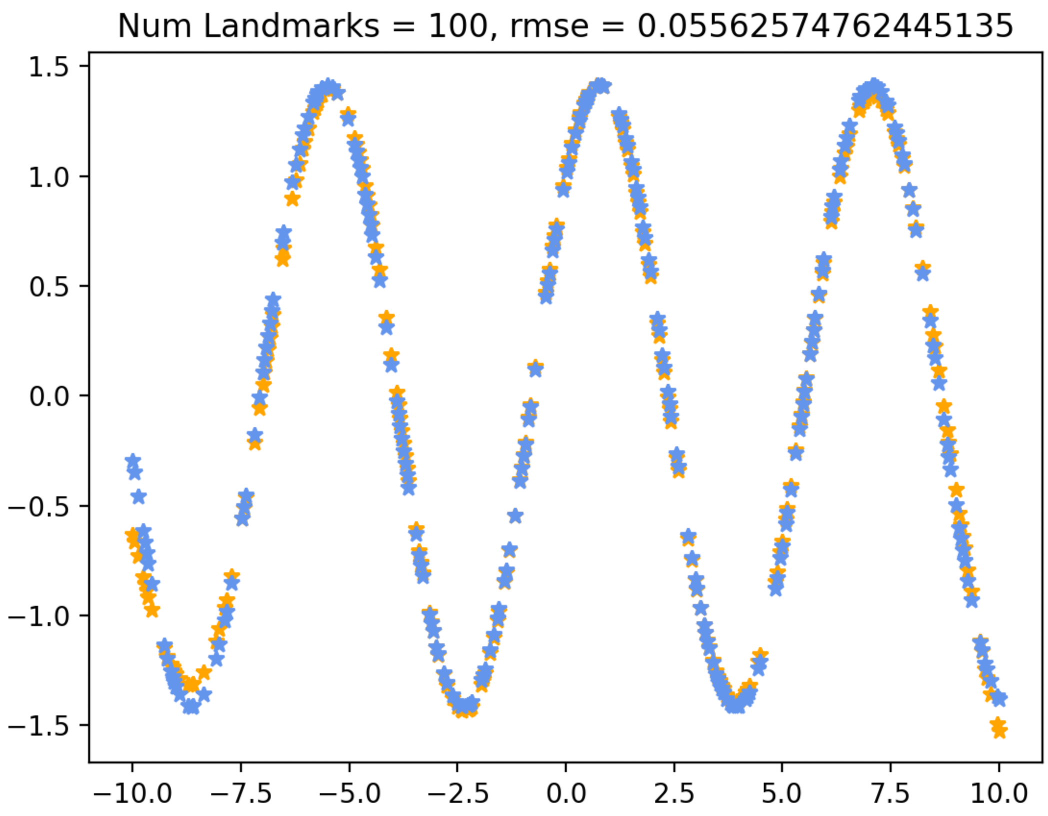
ii) Landmark Ridge regression





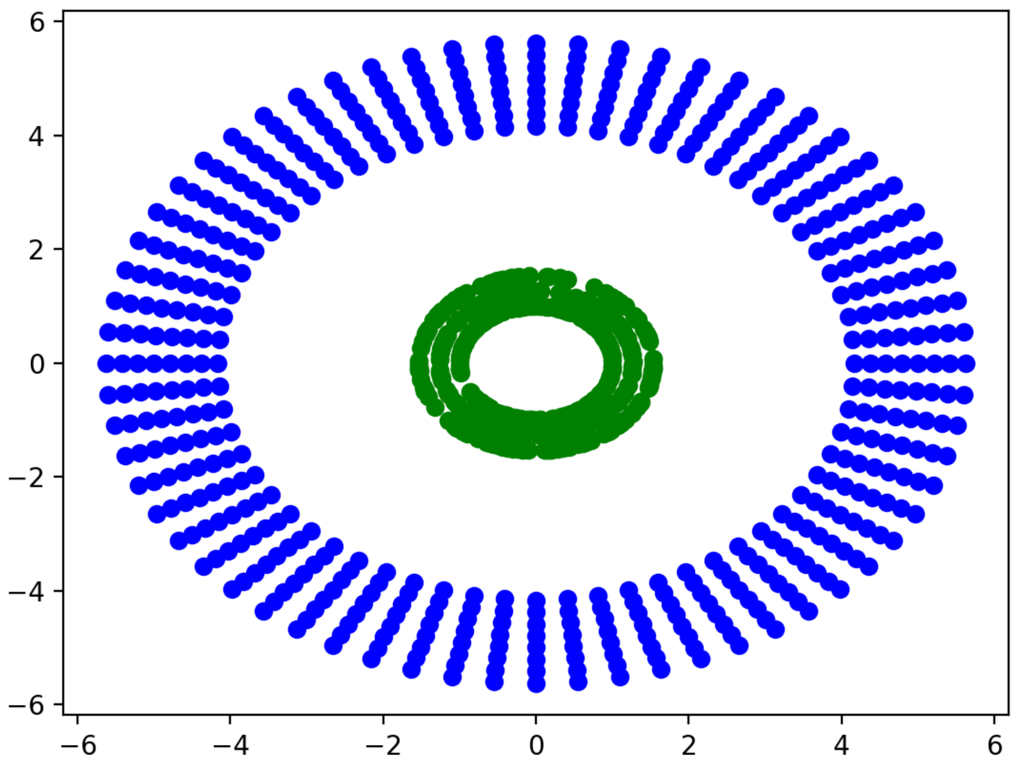

Num Landmarks = 20, rmse = 0.4203903535082229
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−10.0
−7.5
−5.0
−2.5
0.0
2.5
5.0
7.5
10.0


Num Landmarks = 50, rmse = 0.1028019209647163
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−10.0
−7.5
−5.0
−2.5
0.0
2.5
5.0
7.5
10.0

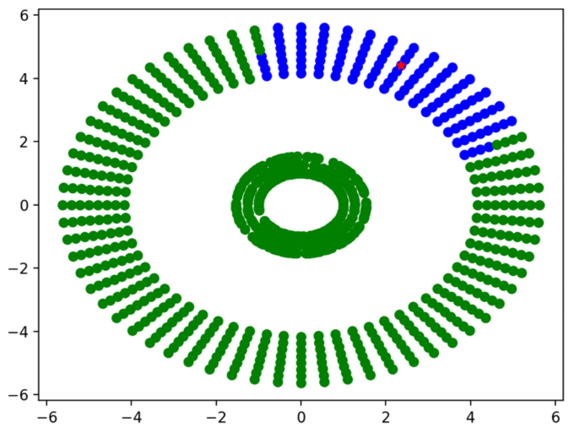

Num Landmarks = 100, rmse = 0.05562574762445135
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−10.0
−7.5
−5.0
−2.5
0.0
2.5
5.0
7.5
10.0

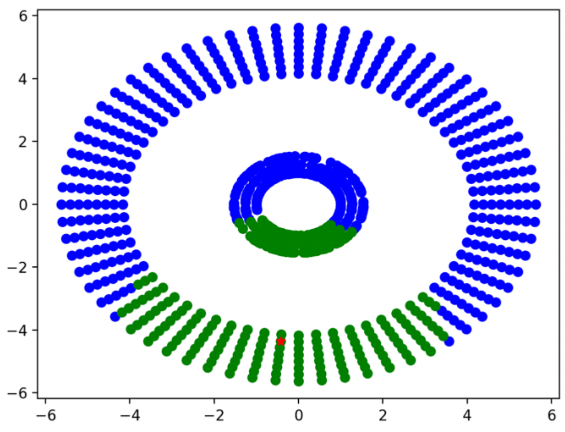
Part2:

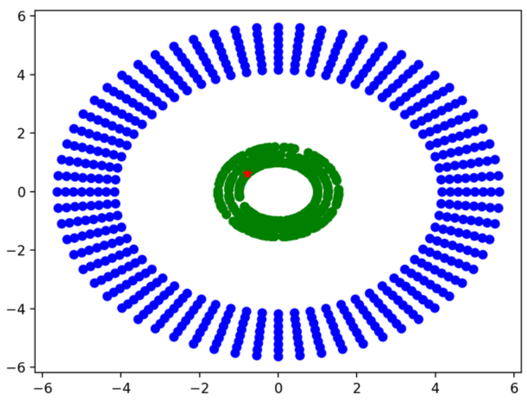
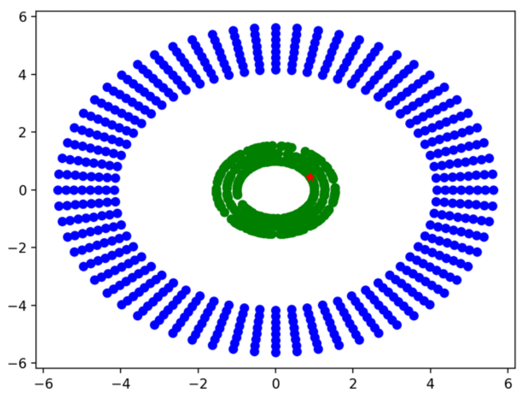
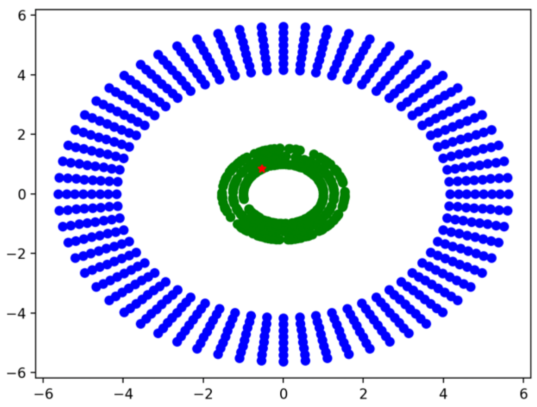
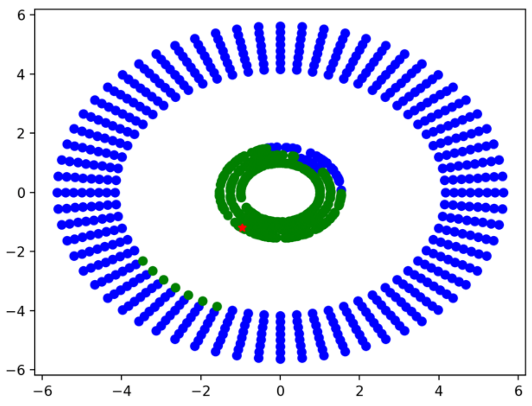
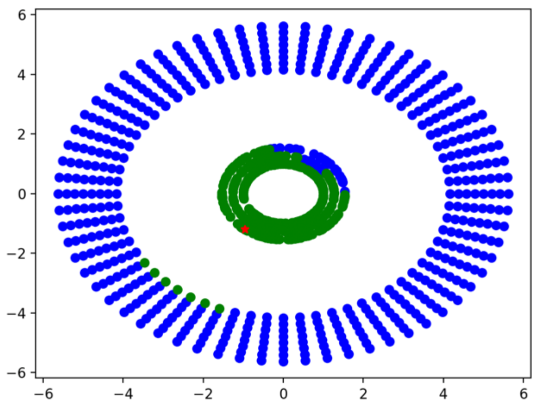
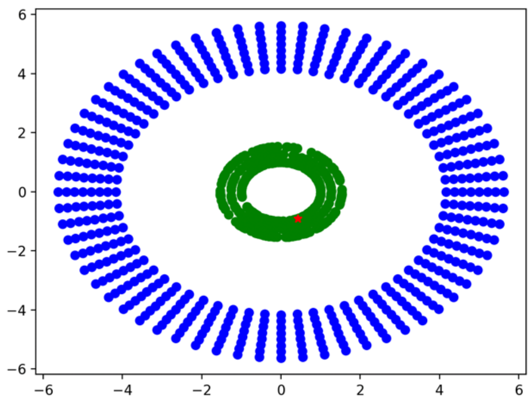
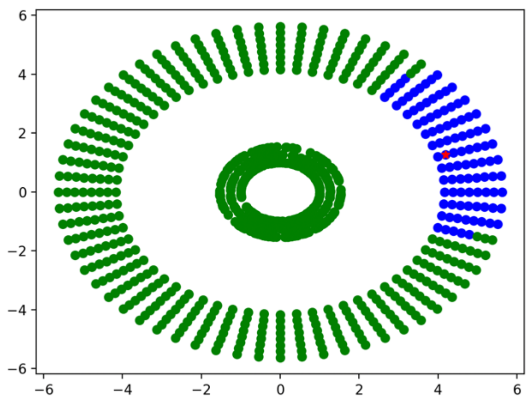
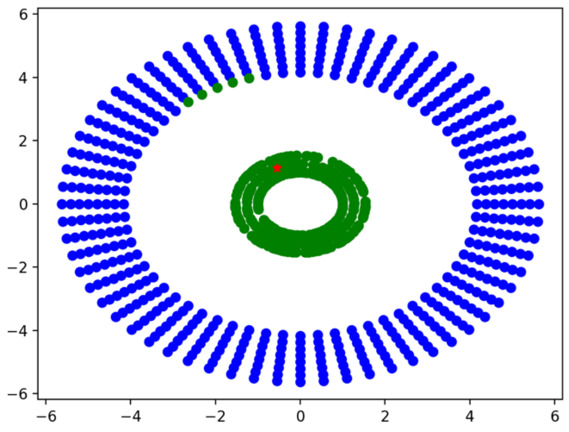
i) Hand-crafted Features:



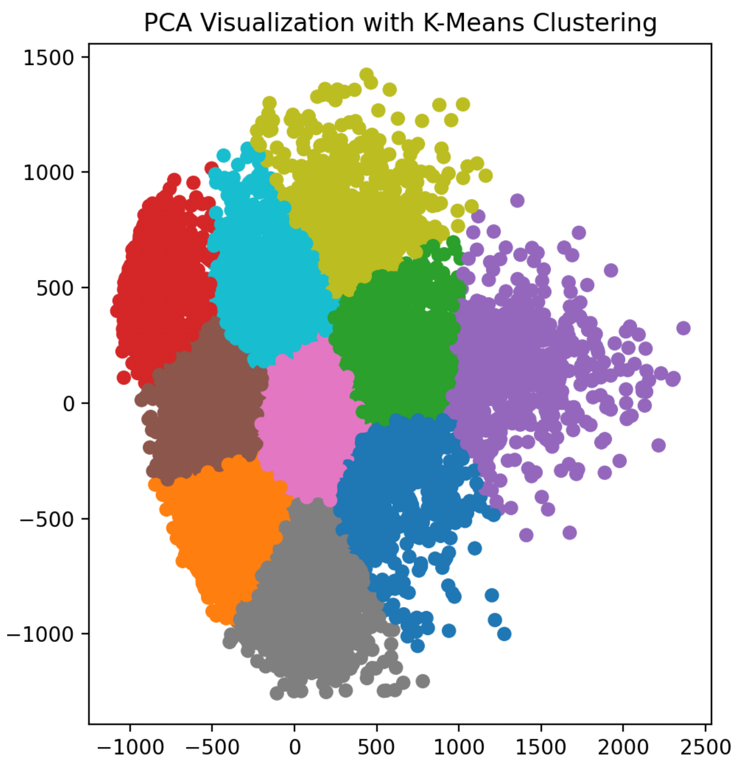
ii) Using Kernels:

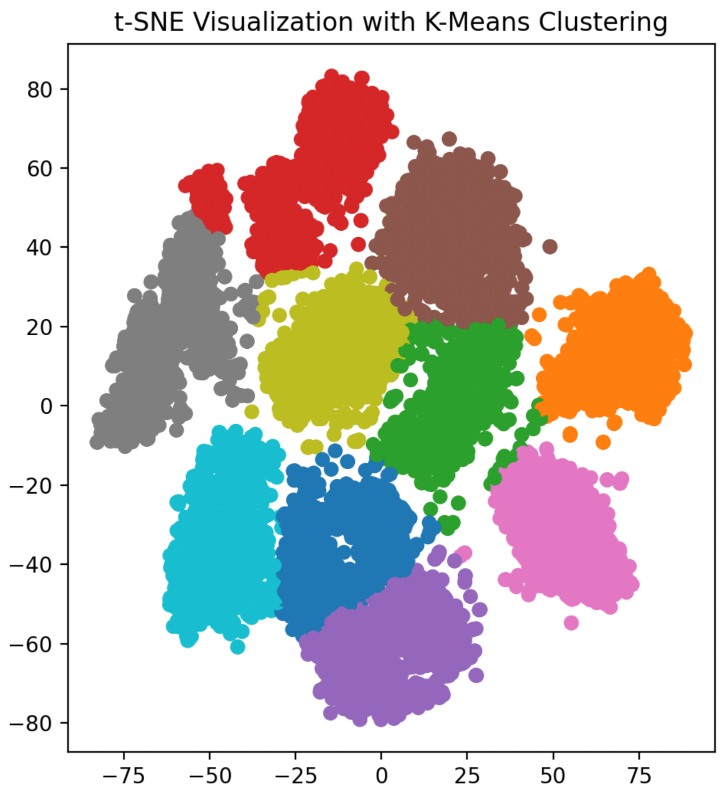





6
4
2
0
−2
−4
−6
−6 −4 −2 0 2 4 6

6
4
2
0
−2
−4
−6
−6 −4 −2 0 2 4 6

6
4
2
0
−2
−4
−6
−6 −4 −2 0 2 4 6

6
4
2
0
−2
−4
−6
−6 −4 −2 0 2 4 6

6
4
2
0
−2
−4
−6
−6 −4 −2 0 2 4 6

6
4
2
0
−2
−4
−6
−6 −4 −2 0 2 4 6

6
4
2
0
−2
−4
−6
−6 −4 −2 0 2 4 6

6
4
2
0
−2
−4
−6
−6 −4 −2 0 2 4 6

Part 3:


PCA Visualization with K-Means Clustering
1500
1000
500
0
−500
−1000
−1000
−500
0
500
1000
1500
2000
2500


t-SNE Visualization with K-Means Clustering
80
60
40
20
0
−20
−40
−60
−80
−75
−50
−25
0
25
50
75