Note:

- GenAI tools are permitted solely for problems: 3, 4, 5.
- Make sure to set the seed before generating random samples.
- Write both full names on the submission if you are working in pairs.

1. In class we assumed that $\mathbb{E}[\phi(x)] = 0$. If this is not the case, the first step is to apply centering before taking the eigendecomposition of K. Show that this is equivalent to computing the eigendecomposition of,

$$K_c = K - \mathbf{1}_N K - K\mathbf{1}_N + \mathbf{1}_N K \mathbf{1}_N$$

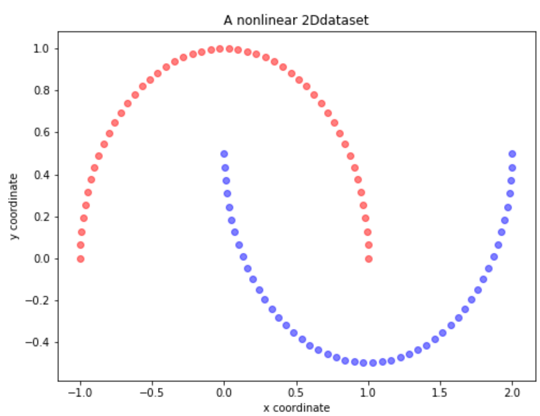
where $\mathbf{1}_N$ is a matrix of $1/N$ in all elements.

2. Based on our derivation of the Nyström extension in class, prove that under the low rank assumption, we can compute K_{22} via,

$$K_{22} = K_{21} K_{11}^{-1} K_{12}$$

3. Coding problem: For the two moons dataset and for the circles dataset, find a parameter setting for which kernel PCA successfully separates the data into two clusters and a parameter setting for which it does not. Explain in 2-3 sentences why it fails.

   - For two moons, in Python use sklearn.datasets.make_moons with 100 samples or generate the data yourself: upper half of a circle with radius 1 whose center is at (0,0) and lower half of a circle with radius 1 whose center is at (1,0.5).



   - For the circles dataset, generate 200 randomly sampled points along a circle of radius 1 and 200 randomly sampled points along a circle of radius 0.25. Add Gaussian noise of std=0.1 to the locations of the points.

4. Coding problem: Generate 2000 points from the two moons dataset.

   - Randomly sample 500 points and perform kernel PCA for the full dataset using the Nyström extension.

- Randomly sample 500 points only from the left moon and perform kernel PCA for the full dataset using the Nyström extension.

Plot the first components for both results and explain.

5. Real data often has anomalous points. Here we investigate the performance of PCA on data containing a single anomaly.

   Generate 25 points randomly distributed along the line passing through $(0,0)$ and $(1,1)$. Use the following steps:

   - Generate a sample $\{u_1,\dots,u_{25}\}$ of uniformly distributed points in $[-1,1]$.
   - Plug the above values in: $X(u_i) = (u_i\cos(\pi/4), u_i\sin(\pi/4))$ to prepare the data $\{x_1,\dots,x_{25}\} \subseteq \mathbb{R}^2$.
   - Finally, add Gaussian noise with std=0.1 to each point.

   (a) Compute and plot the top PC direction of the data.

   (b) Append a *single* anomalous point $(-5,1)$ to the data.

   (c) Again compute and plot the top PC direction of the new data. Visually compare the new PC direction with the old one by plotting both on the same figure.

   (d) Suggest some strategies to resolve the issue (no implementation is required).

   A prospective project would to be investigate L1-norm formulation of PCA which is robust to the anomalies in the data. Specifically, the task would be to develop a computationally efficient algorithm to approximately solve the L1-norm PCA (read more here: `https://en.wikipedia.org/wiki/L1-norm_principal_component_analysis`).