Problem 1

1.

a) Centering or mean removal: centering means making the mean of each variable equal to zero. Centering is an important pre-processing step because it ensures that the resulting components are only looking at the variance within the dataset, and not capturing the overall mean of the dataset as an important variable(dimension). How to apply? Subtract each column of X by mean.

b) Scaling of a feature to a range [a, b]: Feature Scaling, a pre-processing technique that handles cases where our ML models require scaled features for optimal results. feature scaling is important since we are interested in the components that maximize the variance and therefore, we need to ensure that we are comparing apples to apples. How to apply: $x' = a + \frac{(x-\min(x))(b-a)}{\max(x)-\min(x)}$

c) Standardization: all features have mean 0 and unit variance. The reason why it is critical to perform standardization prior to PCA, is that the latter is quite sensitive regarding the variances of the initial variables. How to apply: $x^d = \frac{x_d - \mu_d}{\sigma_d}$

d) Normalization (between 0 and 1): “Normalizing” a vector most often means dividing by a norm of the vector. Normalization is important in PCA since it is a variance maximizing exercise. Xnormalized = (x - xminimum) / range of x

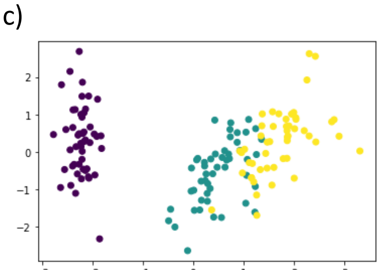
2.

a) [0.92461872]
[0.92461872 0.05306648]
[0.92461872 0.05306648 0.01710261]
[0.92461872 0.05306648 0.01710261 0.00521218]

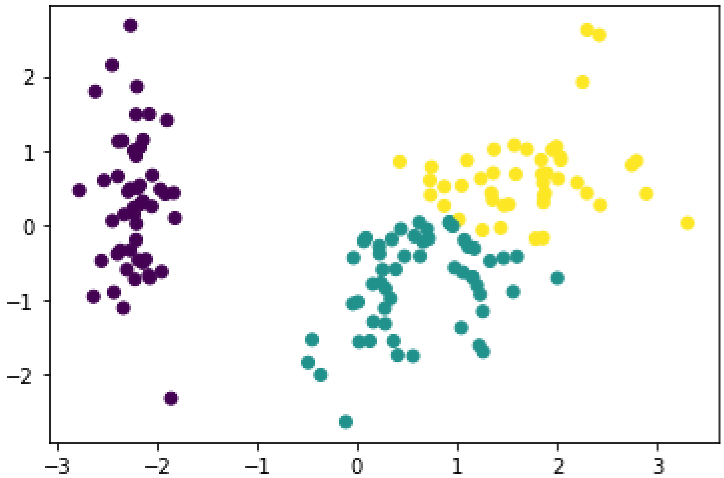
b) [-1.69031455e-15 -1.84297022e-15 -1.69864123e-15 -1.40924309e-15]
[1. 1. 1. 1.]
[0.72962445]
[0.72962445 0.22850762]
[0.72962445 0.22850762 0.03668922]
[0.72962445 0.22850762 0.03668922 0.00517871]

The variance retained when k = 1 become smaller and the variance retained when k =2, 3 become bigger. The difference between k= 2, 3, 4 become more significant. If you don't standardize your data first, these eigenvectors will be all different lengths. Then the eigenspace of the covariance matrix will be "stretched", leading to similarly "stretched" projections.

c)



d) Less overlap in kmeans++ between the clusters



2.

(a) mean(x) = 1

mean(y) = 1.8

$$\begin{bmatrix} 0 & 0.2 \\ -2 & -0.8 \\ -1 & -0.8 \\ 1 & 2.2 \\ 2 & -0.8 \end{bmatrix}$$

(b) $a=0$

$b=1$

$$\begin{bmatrix} 0+\frac{(1-(-1))(1-0)}{3-(-1)} & 0+\frac{(2-1)(1-0)}{4-1} \\ 0+\frac{(-1-(-1))(1-0)}{3-(-1)} & 0+\frac{(1-1)(1-0)}{4-1} \\ 0+\frac{(0-(-1))(1-0)}{3-(-1)} & 0+\frac{(1-1)(1-0)}{4-1} \\ 0+\frac{(2-(-1))(1-0)}{3-(-1)} & 0+\frac{(4-1)(1-0)}{4-1} \\ 0+\frac{(3-(-1))(1-0)}{3-(-1)} & 0+\frac{(1-1)(1-0)}{4-1} \end{bmatrix} = \begin{bmatrix} \frac{1}{2} & \frac{1}{3} \\ 0 & 0 \\ \frac{1}{4} & 0 \\ \frac{3}{4} & 1 \\ 1 & 0 \end{bmatrix}$$

(c)

$$x^d = \frac{x_d - \mu_d}{\sigma_d}$$

$$\sigma_1 = \sqrt{\tfrac{1}{5}[(1-1)^2+(-1-1)^2+(0-1)^2+(2-1)^2+(3-1)^2} = \sqrt{2}$$

$$\sigma_2 = \sqrt{\tfrac{1}{5}[(2-1.8)^2+(1-1.8)^2+(1-1.8)^2+(4-1.8)^2+(1-1.8)^2} = \frac{\sqrt{34}}{5}$$

$$\begin{bmatrix} \frac{1-1}{\sqrt{2}} & \frac{(2-1.8)5}{\sqrt{34}} \\ \frac{-1-1}{\sqrt{2}} & \frac{(1-1.8)5}{\sqrt{34}} \\ \frac{0-1}{\sqrt{2}} & \frac{(1-1.8)5}{\sqrt{34}} \\ \frac{2-1}{\sqrt{2}} & \frac{(4-1.8)5}{\sqrt{34}} \\ \frac{3-1}{\sqrt{2}} & \frac{(1-1.8)5}{\sqrt{34}} \end{bmatrix} = \begin{bmatrix} 0 & \frac{\sqrt{34}}{34} \\ -\sqrt{2} & -\frac{2\sqrt{34}}{17} \\ -\frac{\sqrt{2}}{2} & -\frac{2\sqrt{34}}{17} \\ \frac{\sqrt{2}}{2} & \frac{11\sqrt{34}}{34} \\ \sqrt{2} & -\frac{2\sqrt{34}}{17} \end{bmatrix}$$

(d)

$$\begin{bmatrix} \frac{1-(-1)}{4} & \frac{2-1}{3} \\ \frac{-1-(-1)}{4} & \frac{1-1}{3} \\ \frac{0-(-1)}{4} & \frac{1-1}{3} \\ \frac{2-(-1)}{4} & \frac{4-1}{3} \\ \frac{3-(-1)}{4} & \frac{1-1}{3} \end{bmatrix} = \begin{bmatrix} \frac{1}{2} & \frac{1}{3} \\ 0 & 0 \\ \frac{1}{4} & 0 \\ \frac{3}{4} & 1 \\ 1 & 0 \end{bmatrix}$$

Problem 3

1.

$$\frac{\partial f}{\partial U_i} = \sum_{j\in[m]:(i,j)\in\Omega} 2(r_{i,j} - U_i^T V_j)(-V_j) + 2\alpha U_i = 0$$

$$\sum_{j\in[m]:(i,j)\in\Omega} 2(r_{i,j} - U_i^T V_j)(V_j) = 2\alpha U_i$$

$$\sum_{j\in[m]:(i,j)\in\Omega} r_{i,j} V_j = (\alpha I + \sum_{j\in[m]:(i,j)\in\Omega} V_j V_j^T) U_i$$

$$U_i = (\alpha I + \sum_{j\in[m]:(i,j)\in\Omega} V_j V_j^T)^{-1} \sum_{j\in[m]:(i,j)\in\Omega} r_{i,j} V_j$$

$$\frac{\partial f}{\partial V_j} = \sum_{j\in[m]:(i,j)\in\Omega} 2(r_{i,j} - U_i^T V_j)(-U_i) + 2\beta V_j = 0$$

$$\sum_{j\in[m]:(i,j)\in\Omega} r_{i,j} U_i = 2\beta V_j + \sum_{j\in[m]:(i,j)\in\Omega} U_i U_i^T V_j$$

$$V_j = (\sum_{i\in[n]:(i,j)\in\Omega} U_i U_i^T + \beta I)^{-1} \sum_{i\in[n]:(i,j)\in\Omega} r_{i,j} U_i$$

If $U^{(0)}$ and $V^{(0)}$ are initialized to zero, ~~we cannot use~~ Alternating Minimization algorithm will give us zero.

2) In the gradient calculation. we find that $\sum_{i \in [m], (i,j) \in \Omega} V_j V_j^T$ must be invertible, with the full rank of k. Also, when $\beta = 0$. $\sum_{j \in [m] (i,j) \in \Omega} U_i U_i^T$ has to be invertible and the # of elements has to be equal to the rank There needs to be at least k movies in the total, in order for them to be rated by k^{th} users. Thus we will have k movies as well as k users. Users must rate each movie.

3) U_i:

$$\frac{\partial f}{\partial U_i} = \sum_{j \in [m] (i,j) \in \Omega} 2(r_{i,j} - U_i^T V_j)(-V_j) + 2\alpha U_i$$

$$U_i^{(t)} = U_i - \eta_t (2(U_i^T V_j - R_{ij})(V_j) + 2\alpha U_i)$$

V_j:

$$\frac{\partial f}{\partial V_j} = \sum_{i \in [m] (i,j) \in \Omega} 2(r_{i,j} - U_i^T V_j)(-U_i) + 2\beta V_j)$$

$$V_j^{(t)} = V_j - \eta_t (2(U_i^T V_j - r_{i,j}) U_i + 2\beta V_j)$$

Problem 4

1. 1^{st} policy π_1:

$S_0 \to a_1$

$S_1 \to a_0$

$S_2 \to a_0$

$S_3 \to a_0$

2^{nd} policy π_2:

$S_0 \to a_2$

$S_1 \to a_0$

$S_2 \to a_0$

$S_3 \to a_0$

$V_\pi(S_0) = 0$

$V_\pi(S_1) = 0$

$V_\pi(S_2) = 1$

$V_\pi(S_3) = 10$

$$V_\pi(S) = \sum_a \pi(a|s) \sum_{s',r} P(s', r | S, a)[r + \gamma V_\pi(s')]$$

$r_0 = 0$

$r_1 = 0$

$r_2 = 1$

$r_3 = 10$

$= k - 1)$

2)

$$V^*(S_0) = \max_{a \in (a_1, a_2)} 0 + \gamma \sum_{s'} P(s'|s,a) V^*(s') = \gamma \max\{V^*(S_1), V^*(S_2)\}$$

$$V^*(S_1) = \max_{a \in a_0} 0 + \gamma \sum_{s'} P(s'|s,a) V^*(s')$$

$$= \gamma [(1-p) V^* S_1 + p V^*(S_3)]$$

$$V^*(S_2) = \max_{a \in a_0} 1 + \gamma \sum_{s'} P(s'|s,a) V^*(s') = 1 + \gamma [(1-q) V^*(S_0) + q V^*(S_3)]$$

$$V^*(S_3) = \max_{a \in a_0} 10 + \gamma \sum_{s'} (s'|s,a) V^*(s') = 10 + \gamma V^*(S_0)$$

3)

	S_0	S_1	S_2	S_3
π^*	a_1	a_0	a_0	a_0
V^*	14.1846	15.7608	15.6969	22.7661

Problem 5

According to SARSA algorithm, A' and S' are chosen before updating Q function. By contrast, Q learning will update Q function before choosing the next action. Therefore, they are different.