1.

Since u_{m+1} should be orthogonal to u_i for all $i \in [1, M]$

and $\|u_{m+1}\|^2 = 1$, the new Lagrange Equation is:

$$\mathcal{L} = u_{m+1}^T S u_{m+1} + \lambda(1 - u_{m+1}^T u_{m+1}) + \sum_{i=1}^{M} \alpha_i u_{m+1}^T u_i$$

where $\lambda, \alpha_i \neq 0$.

Now we solve this Lagrange Equation: partials w.r.t each var.

① $\frac{\partial \mathcal{L}}{\partial \lambda} = 1 - u_{m+1}^T u_{m+1} = 0 \Rightarrow u_{m+1}^T u_{m+1} = 1$

② $\frac{\partial \mathcal{L}}{\partial \alpha_i} = u_{m+1}^T u_i = 0$, $\underbrace{\frac{\partial \mathcal{L}}{\partial u_{m+1}} = 2Su_{m+1} - 2\lambda u_{m+1} + \sum_{i=1}^{M} \alpha_i u_i}$ ③

multiply both sides by u_j^T

$$2u_j^T S u_{m+1} - \underbrace{2u_j^T \lambda u_{m+1}}_{0 \text{ b/c } u_j^T u_{m+1} = 0 \text{ from ②}} + \underbrace{\sum_{i=1}^{M} u_j^T \alpha_i u_i}_{} = 0$$

$2u_j^T S u_{m+1} = \underbrace{2\lambda_j u_j^T u_{m+1}}$ because $Su_j = \lambda_j u_j$ and $S = S^T$

$= 0$ ②

this, $\sum_{i=1}^{M} u_j^T \alpha_i u_i = 0$

$= \alpha_j$ b/c $u_j^T u_i = 0$ for all $j \neq i$ ② and $u_j^T u_j = 1$ ①

So, $\alpha_j = 0 \Rightarrow 2u_j^T S u_{m+1} = 2u_j^T \lambda u_{m+1}$

$\cancel{2Su_{m+1} = 2u_j^T}$

$$\boxed{Su_{m+1} = \lambda u_{m+1}}$$.

(2)

$$S = \frac{1}{N}\sum_{n=1}^{N}(x_n-\bar{x})(x_n-\bar{x})^T$$

$$\bar{x} = \begin{bmatrix} 2+0+-2+0 \\ 2-2+0+0 \\ 0+2+0+-2 \end{bmatrix} /4 = \vec{0}$$

$$= \frac{1}{N}\sum_{n=1}^{N} x_n x_n^T = \frac{1}{4}\left(\begin{bmatrix}2\\2\\0\end{bmatrix}[2\ \ 2\ \ 0] + \begin{bmatrix}0\\-2\\2\end{bmatrix}[0\ \ -2\ \ 2] + \begin{bmatrix}-2\\0\\0\end{bmatrix}[-2\ \ 0\ \ 0] + \begin{bmatrix}0\\0\\-2\end{bmatrix}[0\ \ 0\ \ -2]\right)$$

$$= \begin{bmatrix} 2 & 1 & 0 \\ 1 & 2 & -1 \\ 0 & -1 & 2 \end{bmatrix}$$

$$\left\|\begin{matrix} 2-\lambda & 1 & 0 \\ 1 & 2-\lambda & -1 \\ 0 & -1 & 2-\lambda \end{matrix}\right\| = 0 \quad \Rightarrow \quad (2-\lambda)\left(4-4\lambda+\lambda^2-1\right) - 1\cdot(2-\lambda) + 0 = 0$$

$$(2-\lambda)(2-4\lambda+\lambda^2) = 0$$

$$\lambda = 2,\ \frac{4\pm\sqrt{16-8}}{2} = 2\pm\sqrt{2}$$

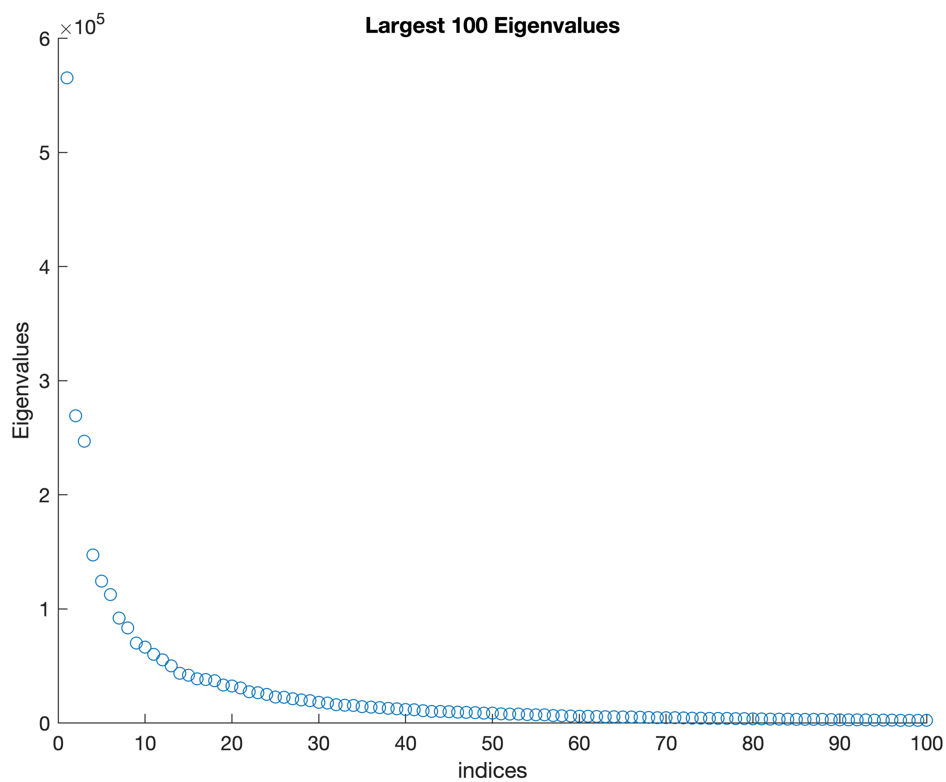
Largest $\lambda = 2+\sqrt{2}$ $\longrightarrow$ $\begin{bmatrix} -\sqrt{2} & 1 & 0 \\ 1 & -\sqrt{2} & -1 \\ 0 & -1 & -\sqrt{2} \end{bmatrix}\begin{bmatrix} a \\ b \\ c \end{bmatrix} = 0$

$$-\sqrt{2}\,a + b = 0,\quad a - b\sqrt{2} - c = 0,\quad -b - c\sqrt{2} = 0$$

Solve, to get $a = b/\sqrt{2}$, $c = -b/\sqrt{2}$, $b = b$. Normalize: $u = \begin{bmatrix} 1/2 \\ \sqrt{2}/2 \\ -1/2 \end{bmatrix}$

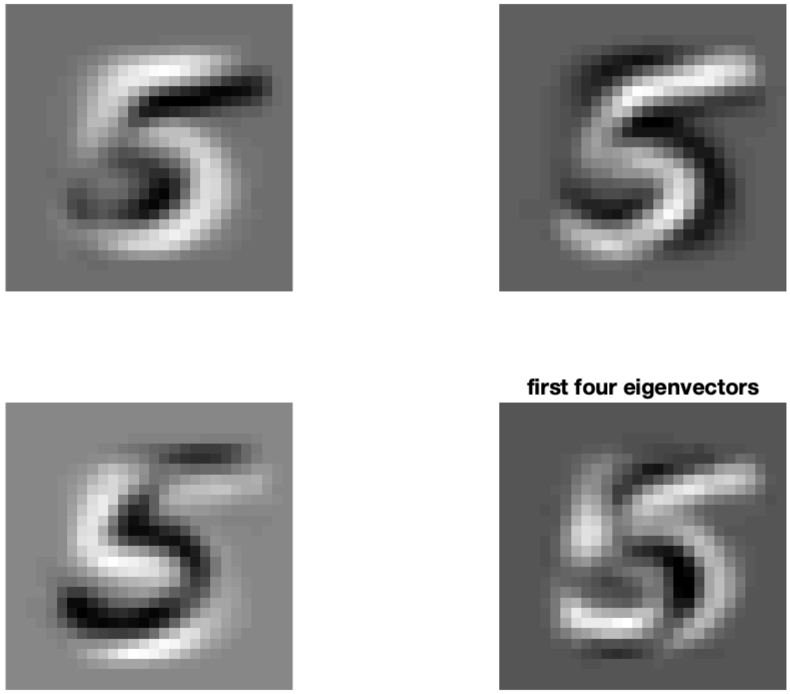
3(a)

```
x = readtable('MNIST5.csv');
x = x{:,:};
covar = cov(x, 1);
[V, D] = eig(covar);
[D, ind] = sort(diag(D), 'descend');
V = V(:,ind);
figure(1);
scatter(1:100, D(1:100));
ylabel('Eigenvalues');
xlabel('indices');
title('Largest 100 Eigenvalues');
```



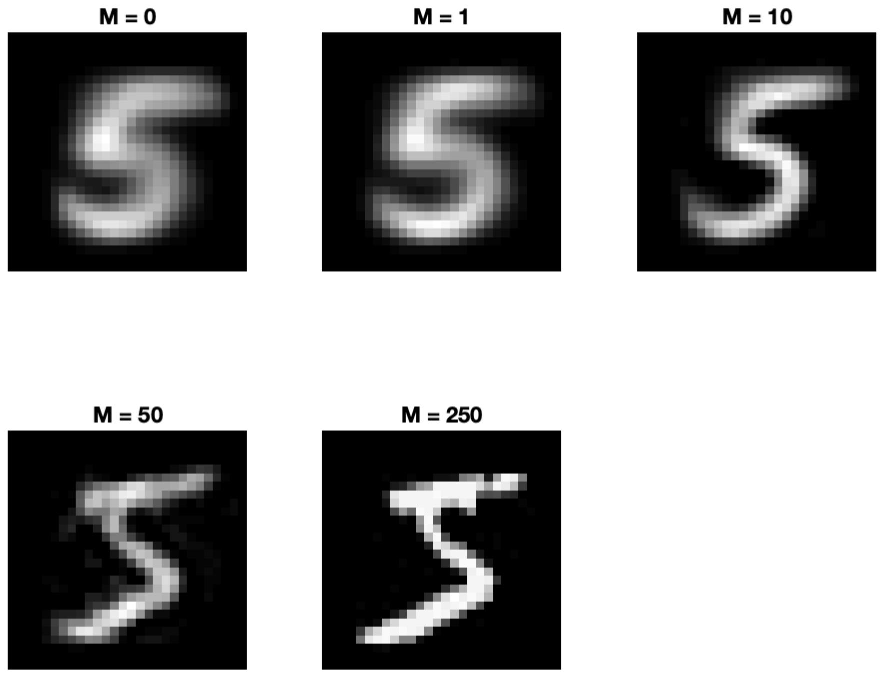
(b)

```
figure(2);
for i = 1:4
    img = cast(rescale(reshape(V(:,i), [28, 28]), 0, 255), 'uint8');
    subplot(2,2,i);
    imshow(img);
end
title('first four eigenvectors');
```



(c)

```
figure(3);
xbar = mean(x, 1);
M = [0;1;10;50;250];
for n = 1:5
    m = M(n);
    xtil = xbar;
    for i = 1:m
        xtil = xtil + (x(1,:) * V(:,i) - xbar * V(:,i)) * V(:,i)';
    end
    subplot(2,3,n),imshow(cast(reshape(xtil, [28, 28]) * 255 / max(xtil), 'uint8'));
    title("M = "+ m);
end
```



M = 250 is very clear. As M increases it looks like the quality of the image increases and the 5 becomes more distinctive.

when M = 0, it is just the mean vector of all of the test cases.

(4)

a. N balls and no replacement

you have to choose #1 at some point.

$P = 0$

b. $P(\text{not } 1) = \frac{N-1}{N}$ $\quad P(N \text{ not } 1\text{'s}) = \left(\frac{N-1}{N}\right)^N \cdot \frac{1}{N} \cdot \binom{N}{1}$

c. $\left(\frac{999}{1000}\right)^{1000} \approx 0.367695 \approx \frac{1}{e}$

$$\lim_{N\to\infty} \left(\frac{N-1}{N}\right)^N = K \qquad \ln(K) = \lim_{N\to\infty} N \cdot \log\left(1-\frac{1}{N}\right)$$

$$= \lim_{N\to\infty} \frac{\log\left(1-\frac{1}{N}\right)}{\frac{1}{N}} = \lim_{a\to 0} \frac{\log(1-a)}{a}$$

by L'Hopital's: $= \lim_{a\to 0} \frac{\frac{-1}{1-a}}{1} = -1$

$$\ln(K) = -1 \Rightarrow K = e^{-1} = \boxed{1/e} \quad \square$$

(5) $\sum_{k=1}^{K} \pi_k = 1 \Rightarrow \lambda\left(\sum_{k=1}^{K} \pi_k - 1\right)$ is the extra lagrangian term.

$$\frac{\partial\left[J + \lambda\left(\sum_{k=1}^{K} \pi_k - 1\right)\right]}{\partial \pi_k} = \sum_{n=1}^{N} \frac{\mathcal{N}(x_n, \mu_k, \Sigma_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(x_n, \mu_j, \Sigma_j)} + \lambda = 0$$

multiply both sides by $\pi_k \Rightarrow \sum_{n=1}^{N} \underbrace{\frac{\pi_k \mathcal{N}(x_n, \mu_k, \Sigma_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(x_n, \mu_j, \Sigma_j)}}_{\gamma(z_{nk})} + \lambda \cdot \pi_k = 0$

$$\sum_{n=1}^{N} \gamma(z_{nk}) + \lambda \pi_k = 0$$

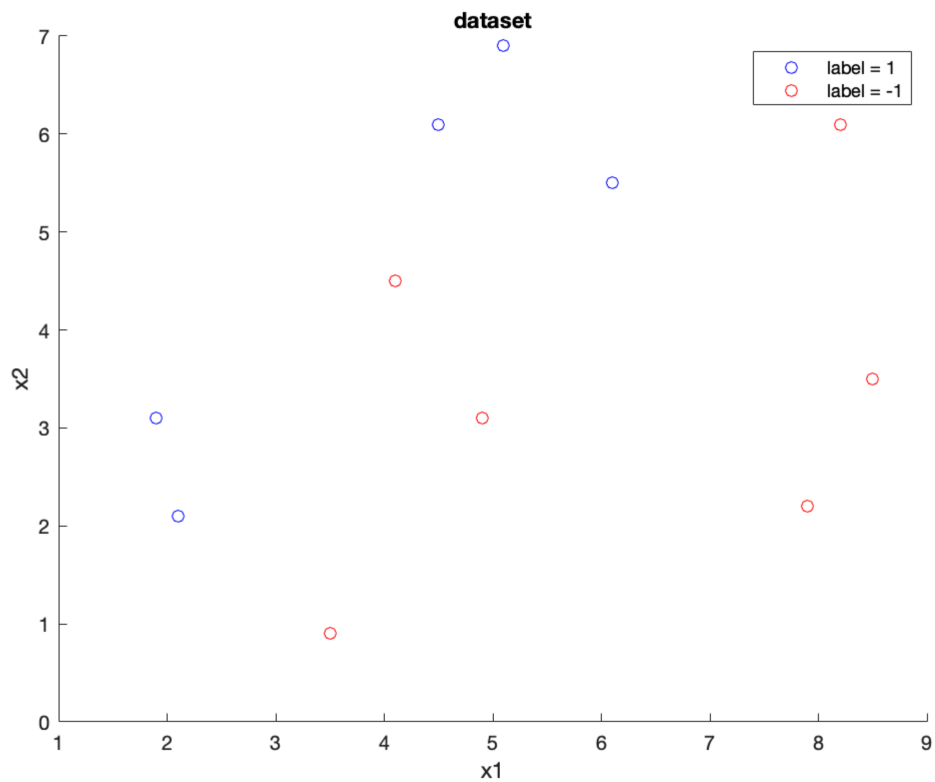
This is true for every π_k → sum over all π_k's: $\sum_{k=1}^{K}\left(\sum_{n=1}^{N} \gamma(z_{nk}) + \lambda \pi_k\right) = 0$

$\sum_{k=1}^{K} \pi_k = 1$, so $N + \lambda = 0$ and $\lambda = -N$ $\qquad = \sum_{n=1}^{N} \sum_{k=1}^{K} \gamma(z_{nk}) = N \cdot 1 = N$

$$\pi_k = \frac{\sum_{n=1}^{N} \gamma(z_{nk})}{N} = \frac{N_k}{N} \quad \square$$

(a)

```
t = readtable('AdaBoost_data.csv');
x = t{:,1:2};
x1 = x(:,1);
x2 = x(:,2);
y = t{:,3};
figure(1);
hold on;
scatter(x1(y == 1), x2(y == 1), 'blue');
scatter(x1(y == -1), x2(y == -1), 'red');
xlabel('x1');
ylabel('x2');
legend('label = 1', 'label = -1');
title('dataset');
```



No, the data is not linearly separable. No, the data cannot be classified using a single layer decision tree.

(b)

```
i = [1 1 2];
s = [-1 -1 1];
t = [Inf Inf Inf];
N = length(y);
d = zeros(K + 1, N);
d(1,:) = 1 / N * ones(1, N);

a = zeros(K, 1);
```

```
for k = 1:3
    xi = x(:,i(k));
    ehatk = Inf;
    yhat = [];
    for tk = floor(min(xi)):ceil(max(xi))
        yh = sign(s(k) * (xi - tk));
        ek = sum(d(k, yh ~= y));
        if ek < ehatk
            ehatk = ek;
            t(k) = tk;
            yhat = yh;
        end
    end
    a(k) = 0.5 * log(1 / ehatk - 1);
    d(k+1,:) = d(k,:) .* exp(-a(k) * y .* yhat)';
    d(k+1,:) = d(k+1,:) / sum(d(k+1,:));
end
dvalues = array2table(d(1:3,:)', 'VariableNames',{'d0', 'd1', 'd2'})
```

dvalues = 11×3 table

	d0	d1	d2
1	0.0909	0.0625	0.0385
2	0.0909	0.0625	0.0385
3	0.0909	0.1667	0.1026
4	0.0909	0.0625	0.1667
5	0.0909	0.0625	0.1667
6	0.0909	0.1667	0.1026
7	0.0909	0.0625	0.1667
8	0.0909	0.1667	0.1026
9	0.0909	0.0625	0.0385
10	0.0909	0.0625	0.0385
11	0.0909	0.0625	0.0385

```
alpha = a'
```

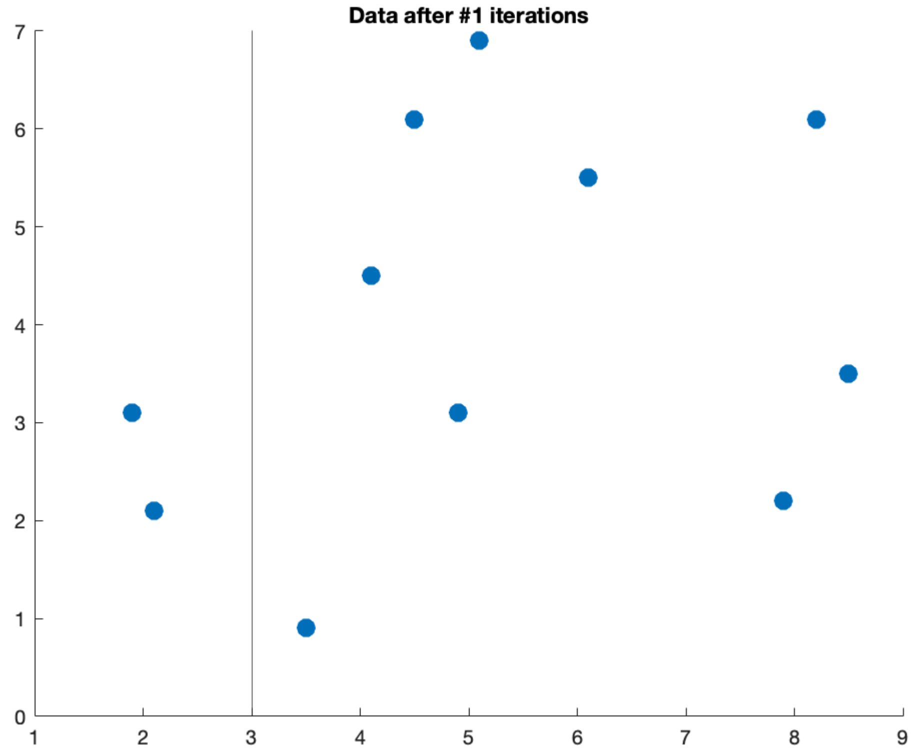
```
alpha = 1x3
    0.4904    0.7332    1.0184
```

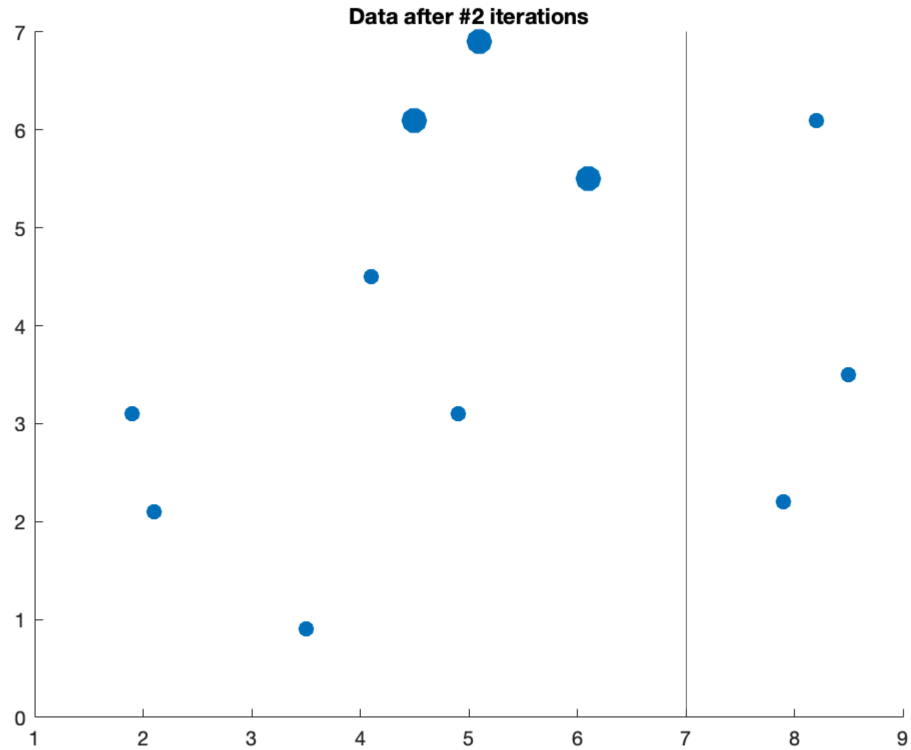
```
tk = t
```

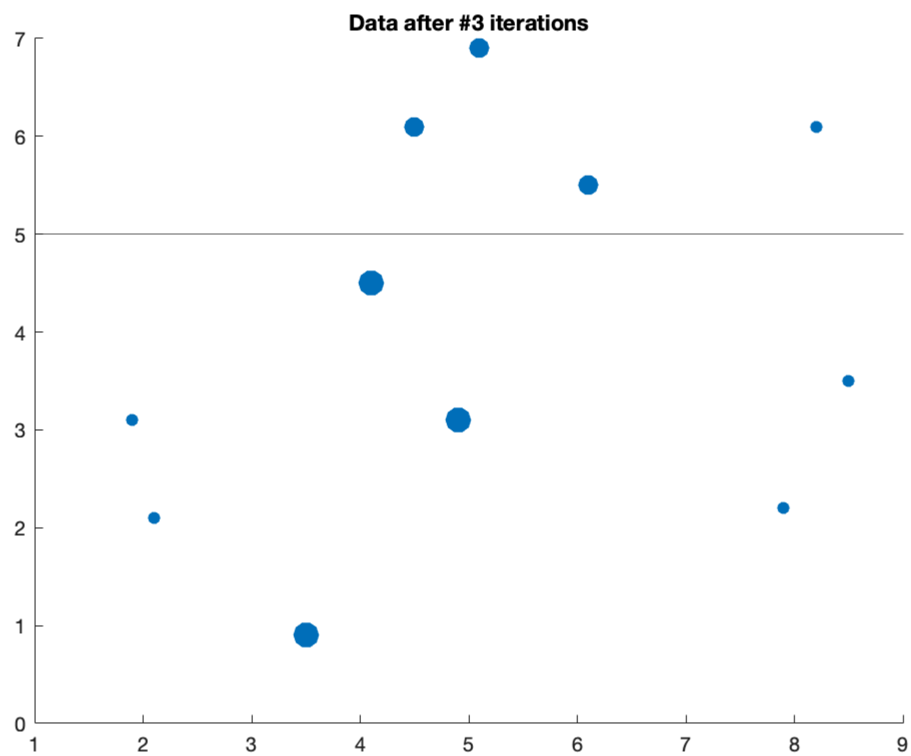
```
tk = 1x3
     3     7     5
```

```
for k = 1:3
    figure(k+1);
    scatter(x1, x2, d(k,:) * 1000, 'filled');
    title(strcat('Data after #', int2str(k), ' iterations'));
    if i(k) == 1
```

```
        xline(t(k));
    else
        yline(t(k));
    end
end
```








```
f = zeros(size(y));
for k = 1:K
    f = f + a(k) * sign(s(k) * (x(:,i(k)) - t(k)));
```

```
end
final_combined_classifier = sign(f)'
```

```
final_combined_classifier = 1×11
     1     1     1    -1    -1     1    -1     1    -1    -1    -1
```

```
acc = 100 * sum(f == y) / N
```

```
acc = 0
```