a)

```
figure(1);
stat = readtable("UCLA_EE_grad_2031.csv");
n = height(stat);
y = stat{:,3};
x = [ones(n,1) stat{:,1:2}]';

a = zeros(n,1);
noK = @(xtest, xi) (xtest' * xi);

for it = 1:1000
    for test = 1:n
        p = 0;
        for i = 1:n
            p = p + a(i) * y(i) * noK(x(:,test), x(:,i));
        end
        if y(test) * p <= 0
            a(test) = a(test) + 1;
        end
    end
end
w = x * (a .* y);
fprintf("no kernel w value:");
```

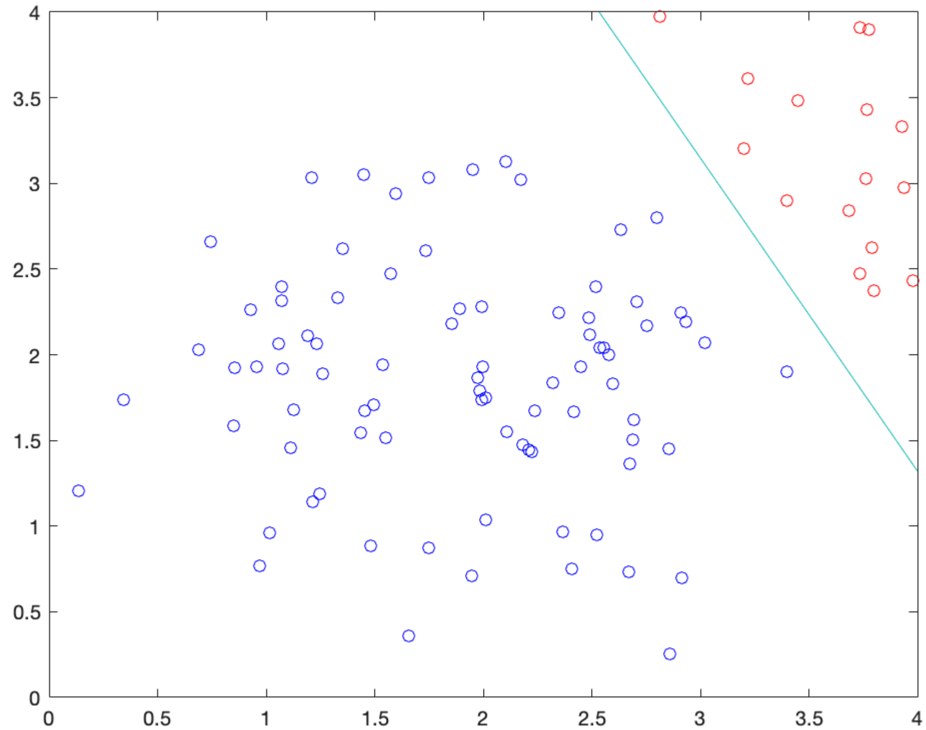
```
no kernel w value:
```

```
w
```

```
w = 3×1
  -28.0000
    5.9282
    3.2447
```

```
xrange = 0:0.1:4;
yrange = 0:0.1:4;
dim = length(xrange);
[x1, x2] = meshgrid(xrange, yrange);
Z = zeros(dim);
for r = 1:dim
    for c = 1:dim
        p = 0;
        for i = 1:n
            p = p + a(i) * y(i) * noK([1; x1(r,c); x2(r,c)], x(:,i));
        end
        Z(r,c) = p;
    end
end
contour(x1, x2, Z, 'LevelList', 0);
hold on;
GPA = x(2,:);
GRE = x(3,:);
scatter(GPA(y == -1), GRE(y == -1), 'blue');
```

```
scatter(GPA(y == 1), GRE(y == 1), 'red');
```




```
miss = 0;
for t = 1:n
    p = 0;
    for i = 1:n
        p = p + a(i) * y(i) * noK(x(:,t), x(:,i));
    end
    if (p * y(t) <= 0)
        miss = miss + 1;
    end
end
acc = 1 - miss/n;
fprintf("no kernel accuracy: %f\n", acc);
```

```
no kernel accuracy: 1.000000
```

b)

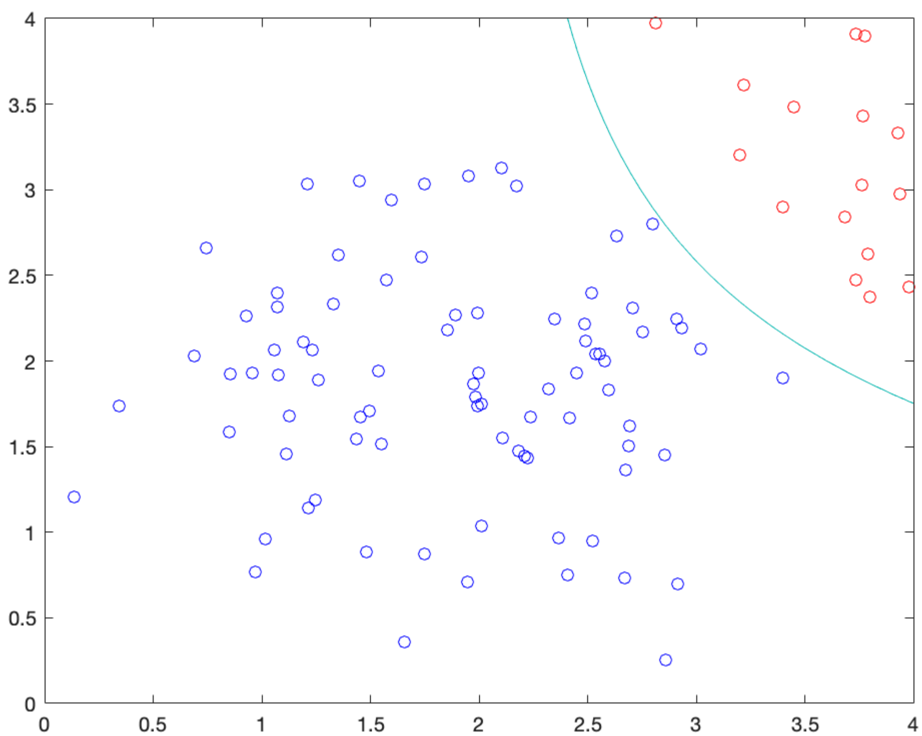
```
figure(2);

a = zeros(n,1);
poK = @(xtest, xi) (1 + xtest' * xi)^2;

for it = 1:1000
    for test = 1:n
        p = 0;
        for i = 1:n
```

```
            p = p + a(i) * y(i) * poK(x(:,test), x(:,i));
        end
        if y(test) * p <= 0
            a(test) = a(test) + 1;
        end
    end
end

xrange = 0:0.1:4;
yrange = 0:0.1:4;
dim = length(xrange);
[x1, x2] = meshgrid(xrange, yrange);
Z = zeros(dim);
for r = 1:dim
    for c = 1:dim
        p = 0;
        for i = 1:n
            p = p + a(i) * y(i) * poK([1; x1(r,c); x2(r,c)], x(:,i));
        end
        Z(r,c) = p;
    end
end
contour(x1, x2, Z, 'LevelList', 0);
hold on;
GPA = x(2,:);
GRE = x(3,:);
scatter(GPA(y == -1), GRE(y == -1), 'blue');
scatter(GPA(y == 1), GRE(y == 1), 'red');
```



```
miss = 0;
for t = 1:n
    p = 0;
    for i = 1:n
        p = p + a(i) * y(i) * poK(x(:,t), x(:,i));
    end
    if (p * y(t) <= 0)
        miss = miss + 1;
    end
end
acc = 1 - miss/n;
fprintf("polynomial kernel accuracy: %f", acc);
```

```
polynomial kernel accuracy: 1.000000
```

(c)

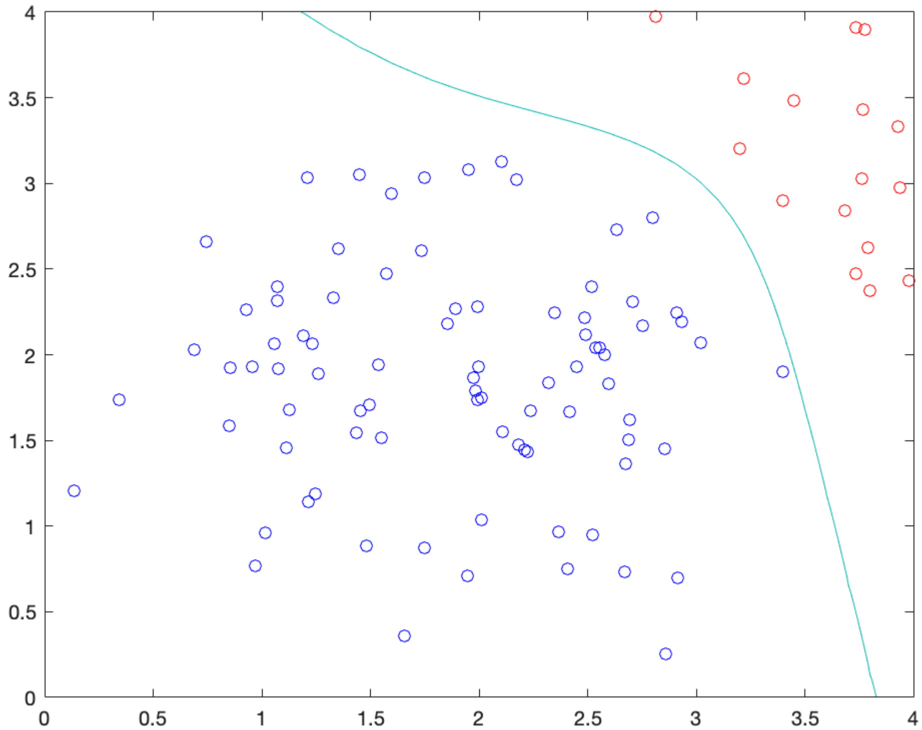
```
figure(3);

a = zeros(n,1);
gaK = @(xtest, xi) exp(-norm(xtest - xi)^2);

for it = 1:1000
    for test = 1:n
        p = 0;
        for i = 1:n
            p = p + a(i) * y(i) * gaK(x(:,test), x(:,i));
        end
        if y(test) * p <= 0
            a(test) = a(test) + 1;
        end
    end
end

xrange = 0:0.1:4;
yrange = 0:0.1:4;
dim = length(xrange);
[x1, x2] = meshgrid(xrange, yrange);
Z = zeros(dim);
for r = 1:dim
    for c = 1:dim
        p = 0;
        for i = 1:n
            p = p + a(i) * y(i) * gaK([1; x1(r,c); x2(r,c)], x(:,i));
        end
        Z(r,c) = p;
    end
end
contour(x1, x2, Z, 'LevelList', 0);
hold on;
GPA = x(2,:);
GRE = x(3,:);
scatter(GPA(y == -1), GRE(y == -1), 'blue');
```

```
scatter(GPA(y == 1), GRE(y == 1), 'red');
```




```
miss = 0;
for t = 1:n
    p = 0;
    for i = 1:n
        p = p + a(i) * y(i) * gaK(x(:,t), x(:,i));
    end
    if (p * y(t) <= 0)
        miss = miss + 1;
    end
end
acc = 1 - miss/n;
fprintf("gaussian kernel accuracy: %f", acc);
```

```
gaussian kernel accuracy: 1.000000
```

(d)

```
figure(4);
hold on;
stat = readtable("UCLA_EE_grad_2030.csv");
n = height(stat);
y = stat{:,3};
x = [ones(n,1) stat{:,1:2}]';
GPA = x(2,:);
GRE = x(3,:);
scatter(GPA(y == -1), GRE(y == -1), 'blue');
```

```
scatter(GPA(y == 1), GRE(y == 1), 'red');


a = zeros(n,1);
gaK = @(xtest, xi) exp(-norm(xtest - xi)^2);

for it = 1:1000
    for test = 1:n
        p = 0;
        for i = 1:n
            p = p + a(i) * y(i) * gaK(x(:,test), x(:,i));
        end
        if y(test) * p <= 0
            a(test) = a(test) + 1;
        end
    end
end

xrange = 0:0.1:4;
yrange = 0:0.1:4;
dim = length(xrange);
[x1, x2] = meshgrid(xrange, yrange);
Z = zeros(dim);
for r = 1:dim
    for c = 1:dim
        p = 0;
        for i = 1:n
            p = p + a(i) * y(i) * gaK([1; x1(r,c); x2(r,c)], x(:,i));
        end
        Z(r,c) = p;
    end
end
contour(x1, x2, Z, 'LevelList', 0, 'LineColor', 'red');

miss = 0;
for t = 1:n
    p = 0;
    for i = 1:n
        p = p + a(i) * y(i) * gaK(x(:,t), x(:,i));
    end
    if (p * y(t) <= 0)
        miss = miss + 1;
    end
end
acc = 1 - miss/n;
fprintf("2030 data now!");
```

```
2030 data now!
```

```
fprintf("gaussian sigma=1 kernel accuracy: %f", acc);
```

```
gaussian sigma=1 kernel accuracy: 0.980000
```

```


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

```
gaK = @(xtest, xi) exp(-3 * norm(xtest - xi)^2);

for it = 1:1000
    for test = 1:n
        p = 0;
        for i = 1:n
            p = p + a(i) * y(i) * gaK(x(:,test), x(:,i));
        end
        if y(test) * p <= 0
            a(test) = a(test) + 1;
        end
    end
end

xrange = 0:0.1:4;
yrange = 0:0.1:4;
dim = length(xrange);
[x1, x2] = meshgrid(xrange, yrange);
Z = zeros(dim);
for r = 1:dim
    for c = 1:dim
        p = 0;
        for i = 1:n
            p = p + a(i) * y(i) * gaK([1; x1(r,c); x2(r,c)], x(:,i));
        end
        Z(r,c) = p;
    end
end
contour(x1, x2, Z, 'LevelList', 0, 'LineColor', 'blue');

miss = 0;
for t = 1:n
    p = 0;
    for i = 1:n
        p = p + a(i) * y(i) * gaK(x(:,t), x(:,i));
    end
    if (p * y(t) <= 0)
        miss = miss + 1;
    end
end
acc = 1 - miss/n;
fprintf("gaussian sigma=3 kernel accuracy: %f", acc);
```

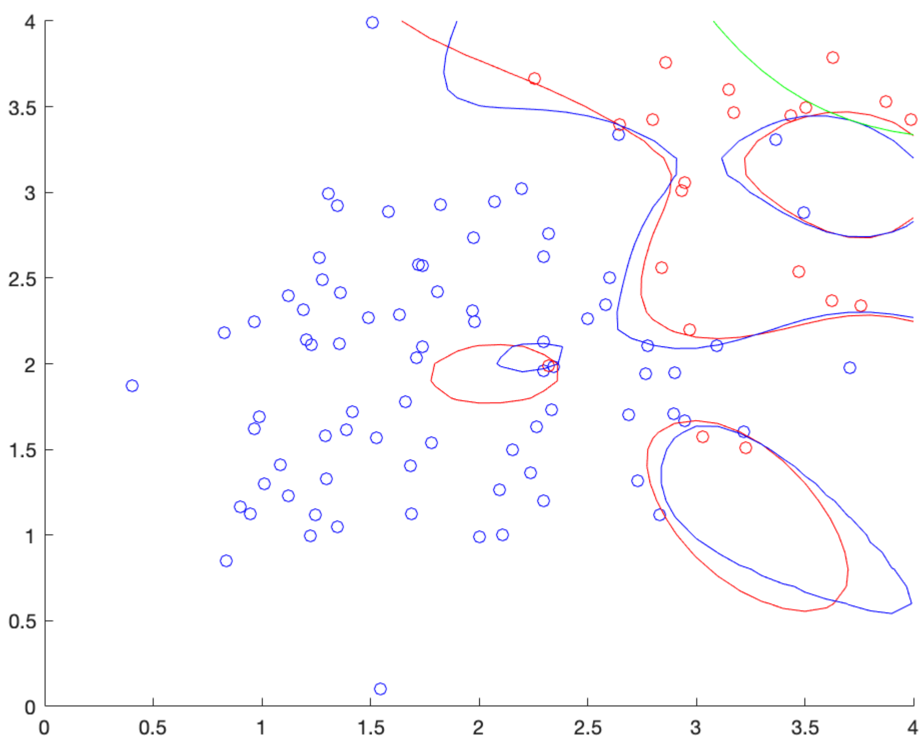
```
gaussian sigma=3 kernel accuracy: 1.000000
```

```
a = zeros(n,1);
poK = @(xtest, xi) (1 + xtest' * xi)^2;

for it = 1:1000
    for test = 1:n
        p = 0;
        for i = 1:n
            p = p + a(i) * y(i) * poK(x(:,test), x(:,i));
```

```
        end
        if y(test) * p <= 0
            a(test) = a(test) + 1;
        end
    end
end

xrange = 0:0.1:4;
yrange = 0:0.1:4;
dim = length(xrange);
[x1, x2] = meshgrid(xrange, yrange);
Z = zeros(dim);
for r = 1:dim
    for c = 1:dim
        p = 0;
        for i = 1:n
            p = p + a(i) * y(i) * poK([1; x1(r,c); x2(r,c)], x(:,i));
        end
        Z(r,c) = p;
    end
end
contour(x1, x2, Z, 'LevelList', 0, 'LineColor', 'green');
```




```
miss = 0;
for t = 1:n
    p = 0;
    for i = 1:n
```

```
        p = p + a(i) * y(i) * poK(x(:,t), x(:,i));
    end
    if (p * y(t) <= 0)
        miss = miss + 1;
    end
end
acc = 1 - miss/n;
fprintf("polynomial degree=2 kernel accuracy: %f", acc);
```

```
polynomial degree=2 kernel accuracy: 0.820000
```

(d) The gaussian kernel with sigma equal to 1 had a 98 percent accuracy. The gaussian kernel with sigma equal to 3 had a 100 percent accuracy. It seems that if there is a higher sigma the accuracy is higher. The polynomial kernel was trash. It got 0.82 percent and only had 3 points in one of its classes. Only the gaussian kernel could get 100% because there were thingies inside the blue class that were part of the red class. (Was not linearly separable).

2) (a) $P(A|C=0) = \cancel{0.8}$ $\frac{0.28}{0.4} = 0.7$

$P(B|C=0) = \frac{0.08}{0.32} = \frac{\cancel{0.25}}{\cancel{0.320}}$ 0.08 $\frac{0.08}{0.400} = \cancel{0.8}$ 0.2

0.420	0.096	0.024	0.27	0.03
0.580	0.224	0.056	0.27	0.03
	0.320	0.080	0.54	0.06

$P(A, B|C=0) = \frac{0.056}{0.40} = 0.14$

$P(A=0|C=0) = 1-0.7 = 0.3$

$P(B=0|C=0) = 1-0.2 = 0.8$

$P(A=1, B=0|C=0) = 0.224/0.4 = 0.56$

$P(A=0, B=1|C=0) = 0.06$

$P(A=0, B=0|C=0) = 0.024$

(b) $P(A|C=1) = \frac{0.27+0.03}{0.6} = 0.5$

$P(A=0|C=1) = 0.5$

$P(B=0|C=1) = 0.9$

$P(B|C=1) = \frac{0.06}{0.60} = 0.1$

$P(A=1, B=0|C=1) = 0.27/0.60 = 0.45$

$P(A=0, B=1|C=1) = 0.03/0.60 = 0.05$

$P(A=0, B=0|C=1) = 0.45$

$P(A,B|C=1)$ $\frac{0.03}{0.06} = 0.05$

(c) Yes conditionally independent b/c $P(A|C=0) \cdot P(B|C=0) = P(A,B|C=0)$

$0.7 * 0.2 = 0.14$ for all cases.

C=1 " " "

$0.5 \cdot 0.1 = 0.05$

is $P(A|C) \cdot P(B|C) = P(A,B|C)$ independent

[yes conditionally indep.]

(d) $P(A) = 0.580$ A=1 $P(A=0) = 0.42$

~~P(B) = 0.420~~ $P(B) = 0.14$ B=1 $P(B=0) = 0.86$

$P(A, B) = 0.086$ A=B=1

$P(A=0, B=0) = 0.366$

$P(A=1, B=0) = 0.494$

$P(A=0, B=1) = 0.054$

(e) $P(A) \cdot P(B) = P(A,B)$

$.58 \cdot .14 \neq .086$

NOT INDEPENDENT

does not hold for all combos of A, B

(3) (a). $P(G) = P(G=0) = 2/8 = 1/4$ $\quad P(G=1) = 3/4$

$P(X|G) \to P(\cancel{\text{tot } G}) =$

$P(O=0|G=0) = 0$ $\quad P(A=0|G=0) = 1$ $\quad P(C=0|G=0) = 1/2$

$P(O=1|G=0) = 1$ $\quad P(A=0|G=1) = 1/6$ $\quad C=0|G=1 = 1/2$

$P(O=0|G=1) = 0.5$ $\quad P(A=1|G=0) = 0$ $\quad C=1|G=0 = 1/2$

$P(O=1|G=1) = 0.5$ $\quad P(A=1|G=1) = 5/6$ $\quad C=1|G=1 = 1/2$

$P(B=0|G=0) = 0$

$B=1|G=0 = 1$

$B=0|G=1 = 2/3$

$B=1|G=1 = 1/3$

counting in graph

(b). $P(G=0) \cdot P(O=0|G=0) \cdot P(B=1|G=0)\, P(C=0|G=0) \cdot P(A=1|G=0)$

$= 1/4 \cdot 0 \cdot 1 \cdot \frac{1}{2} \cdot 0 = 0$

$P(G=1) \cdot P(O=0|G=1) \cdot P(B=1|G=1)\, P(C=0|G=1)\, P(A=1|G=1)$

$= 3/4 \cdot \frac{1}{2} \cdot \frac{1}{3} \cdot \frac{1}{2} \cdot \frac{5}{6} = \frac{5}{96}$ $\quad G=1 > G=0$

$\boxed{G=1}$ for sample 9

$P(G=0)\, P(O=1|G=0)\, P(B=1|G=0)\, P(C=1|G=0)\, P(A=1|G=0)$

$= 1 \cdot 1 \cdot \frac{1}{2} \cdot 0 \cdot \frac{1}{4} = 0$

$P(G=1)\, P(O=1|G=1)\, P(B=1|G=1)\, P(C=1|G=1)\, P(A=1|G=1)$

$= \frac{3}{4} \cdot \frac{1}{2} \cdot \frac{1}{3} \cdot \frac{1}{2} \cdot \frac{5}{6} = \frac{5}{96}$

$G=1 > G=0$

sample 10 $\boxed{G=1}$

(c). $P(O=0|G=0) = 1/4$ $\quad P(A=0|G=0) = 3/4$ $\quad P(B=0|G=0) = 1/4$

$P(O=1|G=0) = 3/4$ $\quad P(A=1|G=0) = 1/4$ $\quad P(B=1|G=0) = 3/4$

$P(O=0|G=1) = 4/8$ $\quad P(A=0|G=1) = 2/8$ $\quad P(B=0|G=1) = 5/8$

$P(O=1|G=1) = 4/8$ $\quad P(A=1|G=1) = 6/8$ $\quad P(B=1|G=1) = 3/8$

$P(C=0|G=0) = 1/2$

$P(C=1|G=0) = 1/2$

$P(C=0|G=1) = 1/2$

$P(C=1|G=1) = 1/2$

(d) $P(G=0)\,P(O=0|G=0)\,P(B=1|G=0)\,P(C=0|G=0)\,P(A=1|G=0)$

$= 2/8 \cdot 1/4 \cdot 3/4 \cdot 1/2 \cdot 1/4 = 0.00586$

$P(G=1)\,P(O=0|G=1)\,P(B=1|G=1)\,P(C=0|G=1)\,P(A=1|G=1)$

$= 1/2 \cdot 3/8 \cdot 1/2 \cdot 3/4 \cdot 6/8 = 0.0527$

For sample 9: $G_1 > G_0 \Rightarrow \boxed{G=1}$

$P(G=0)\,P(O=1|G=0)\,P(B=1|G=0)\,P(C=1|G=0)\,P(A=1|G=0)$

$= 2/8 \cdot 3/4 \cdot 3/4 \cdot 1/2 \cdot 1/4 = 0.0176$

$P(G=1)\,P(O=1|G=1)\,P(B=1|G=1)\,P(C=1|G=1)\,P(A=1|G=1)$

$= 1/2 \cdot 3/8 \cdot 1/2 \cdot 3/4 \cdot 6/8 = 0.0527$

For sample 10: $G_1 > G_0 \Rightarrow \boxed{G=1}$

(4) $\cancel{P(x^1, x^2 \dots y^1, y^2 \dots y^m) = \prod_{k=1}^{m} P(x^k, y^k)}$

$\cancel{= \prod_{k=1}^{m} P(y^{(k)}) \cdot P(x^{(k)}|y^{(k)}) = \prod_{k=1}^{m} P(y^{(k)}) \cdot P}$

$=$

Problem 4

(a) $$\prod_{i=1}^{m} P(x^{(i)}, y^{(i)}) = \prod_{i=1}^{m} P(y^{(i)}) \cdot P(x^{(i)} | y^{(i)})$$

$$= \prod_{i=1}^{m} \left[\theta_0^{1(y^{(i)}=0)} (1-\theta_0)^{1(y^{(i)}=1)} \prod_{j=1}^{n} \left(\prod_{k=1}^{s} \theta_{j,k|y=0}^{1[x_j^{(i)}=k, y^{(i)}=0]} \cdot \left(1 - \sum_{k=1}^{s-1} \theta_{j,k|y=0}\right)^{1[x_j^{(i)}=s, y^{(i)}=0]} \cdot \theta_{j,k|y=1}^{1[x_j^{(i)}=k, y^{(i)}=1]} \left(1 - \sum_{k=1}^{s-1} \theta_{j,k|y=1}\right)^{1[x_j^{(i)}=s, y^{(i)}=1]} \right) \right]$$

(b) $$\log((a)) = \sum_{i=1}^{m} 1[y^{(i)}=1] \log(1-\theta_0) + 1[y^{(i)}=0] \log\theta_0$$
$$+ \sum_{j=1}^{n} \sum_{k=1}^{s} 1[x_j^{(i)}=k, y^{(i)}=0] \log \theta_{j,k|y=0}$$
$$+ 1[x_j^{(i)}=s, y^{(i)}=0] \log\left(1 - \sum_{k=1}^{j-1} \theta_{j,k|y=0}\right)$$
$$+ 1[x_j^{(i)}=k, y^{(i)}=1] \log \theta_{j,k|y=1}$$
$$+ 1[x_j^{(i)}=s, y^{(i)}=1] \log\left(1 - \sum_{k=1}^{s-1} \theta_{j,k|y=1}\right)$$

$$\frac{\partial}{\partial\theta_0} = \sum_{i=1}^{m} 1[y^{(i)}=1] \cdot \log(1-\theta_0) + 1[y^{(i)}=0] \log\theta_0$$

$N_{k,0} = 1[x_j^{(i)}=k, y^{(i)}=0]$ $N_{k,1} = 1[x_j^{(i)}=k, y^{(i)}=1]$ $N_{s,0} = 1[x_j^{(i)}=s, y^{(i)}=0]$

$N_{s,1} = 1[x_j^{(i)}=s, y^{(i)}=1]$

$$\frac{-N_1}{1-\theta_0} + \frac{N_0}{\theta_0} = 0 \quad \text{where } N_1 = \sum_{i=1}^{m} 1[y^{(i)}=1] \quad N_0 = \sum_{i=1}^{m} 1[y^{(i)}=0]$$

$$\boxed{\hat{\theta}_0 = N_0/(N_0+N_1)}$$

$\sum_{j=1}^{n}$ $\sum_{k=1}^{s}$ disappears b/c w.r.t k.

$$\frac{\partial P}{\partial\theta_{j,k|y=0}} = \sum_{i=1}^{m}\sum_{j=1}^{n} \frac{N_{k,0}}{\theta_{j,k|y=0}} + \frac{-N_{s,0}}{1-\sum_{k=1}^{j-1}\theta_{j,k|y=0}} = 0 \Rightarrow \frac{\sum_{i=1}^{m} N_{k,0}}{\theta_{j,k|y=0}} = \frac{\sum_{i=1}^{m} N_{s,0}}{\theta_{j,s|y=0}}$$

$$\rightarrow \boxed{\hat{\theta}_{j,k|y=0} = \theta_{j,s|y=0} \frac{\sum_{i=1}^{m} N_{k,0}}{\sum_{i=1}^{m} N_{s,0}}} \Longrightarrow$$

Wear

$$\boxed{\hat{\theta}_{j,k|y=1} = \theta_{j,s|y=1} \frac{\sum_{i=1}^{m} N_{k,1}}{\sum_{r=1}^{m} N_{s,1}}}$$ by symmetry

$\hat{\theta}_0$ the proportion of all trials whose coin flip land said $y=0$

$\hat{\theta}_{j,k|y=0}$ the proportion to the ratio of # trials where the jth die roll is ~~not the max val~~ k over the # trials whose ~~die~~ roll jth roll is s (the max) all with ~~die~~ coin flip $= 0$.

$\hat{\theta}_{j,k|y=1}$ the same thing except dom flip $= (y=1)$

5.

$$f_{XY}(x,y) = \frac{\exp\left\{ \frac{-1}{2(1-\rho_{XY}^2)} \left[\left(\frac{x-m_1}{\sigma_1}\right)^2 - 2\rho_{XY}\left(\frac{x-m_1}{\sigma_1}\right)\left(\frac{y-m_2}{\sigma_2}\right) + \left(\frac{y-m_2}{\sigma_2}\right)^2 \right] \right\}}{2\pi\sigma_1\sigma_2\sqrt{1-\rho_{XY}^2}}$$

$$\Sigma = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \rightarrow |\Sigma| = ad-bc$$

$$ad-bc = \sigma_1^2\sigma_2^2 - \rho_{XY}^2\sigma_1^2\sigma_2^2$$

$ad =$ $bc =$

$$\boxed{\Sigma = \begin{bmatrix} \sigma_1^2 & \rho_{XY}\sigma_1\sigma_2 \\ \rho_{XY}\sigma_1\sigma_2 & \sigma_2^2 \end{bmatrix}}$$

$$\hookrightarrow \Sigma^{-1} = \boxed{\frac{1}{\sigma_1^2\sigma_2^2 - \rho_{XY}^2\sigma_1^2\sigma_2^2}\begin{bmatrix} \sigma_2^2 & -\rho_{XY}\sigma_1\sigma_2 \\ -\rho_{XY}\sigma_1\sigma_2 & \sigma_1^2 \end{bmatrix}}$$

b/c $$\Sigma^{-1} = \frac{1}{ad-bc}\begin{bmatrix} d & -b \\ -c & a \end{bmatrix}$$

$$z = \begin{bmatrix} x \\ y \end{bmatrix} \quad \mu = \begin{bmatrix} c_1 \\ c_2 \end{bmatrix}$$

$$(z-\mu)^T \Sigma^{-1} (z-\mu)$$

$$= \begin{bmatrix} x-c_1 \\ y-c_2 \end{bmatrix}^T \quad \begin{bmatrix} x-c_1 \\ y-c_2 \end{bmatrix}$$

$$= \frac{1}{(1-\rho_{XY}^2)}\left[\left(\frac{x-c_1}{\sigma_1}\right)^2 - 2\rho_{XY}\frac{(x-c_1)(y-c_2)}{\sigma_1\sigma_2} + \left(\frac{y-c_2}{\sigma_2}\right)^2\right]$$

$$\boxed{\mu = \begin{bmatrix} m_1 \\ m_2 \end{bmatrix}}$$

i) $\cancel{\Sigma_{ij} = \mathrm{cov}[x_i, x_j]} =$

x_i, x_j are independent

$$E(x_i, x_j) = E[x_i] \cdot E[x_j] \leftarrow \iint x_i x_j \, p(x_i, x_j)$$

$$= \int x_i \, p_{xi} \quad \int x_j \, p_{xj}$$

$$\mathrm{cov}[x_i, x_j] = \cancel{E[x_i, x_j]} - \cancel{E[x_i] E[x_j]}$$

$$= 0$$

$$\begin{bmatrix} \sigma_1^2 & 0 & 0 & 0 & 0 \\ & \sigma_2^2 & 0 & 0 & 0 \\ 0 & & \ddots & 0 & 0 \\ & 0 & & & \\ 0 & & 0 & \sigma_N^2 & \end{bmatrix}$$

a)

```
stat = readtable("UCLA_EE_grad_2030.csv");
y = (stat{:,3} + 1)/2;
GPA = stat{:,1};
GRE = stat{:,2};

N = length(y);

Py0 = sum(y == 0) / N
```

```
Py0 = 0.7900
```

```
GPAmu0 = sum(GPA(y == 0)) / sum(y == 0)
```

```
GPAmu0 = 1.8678
```

```
GPAmu1 = sum(GPA(y == 1)) / sum(y == 1)
```

```
GPAmu1 = 3.1637
```

```
GPAvar = 1 / N * (norm(GPA(y == 0) - GPAmu0)^2 + norm(GPA(y == 1) - GPAmu1)^2)
```

```
GPAvar = 0.4457
```

```
GREmu0 = sum(GRE(y == 0)) / sum(y == 0)
```

```
GREmu0 = 1.9673
```

```
GREmu1 = sum(GRE(y == 1)) / sum(y == 1)
```

```
GREmu1 = 2.9590
```

```
GREvar = 1 / N * (norm(GRE(y == 0) - GREmu0)^2 + norm(GRE(y == 1) - GREmu1)^2)
```

```
GREvar = 0.4745
```

b)

```
theta = 1 - Py0;
b1 = (GPAmu1^2 - GPAmu0^2) / (2 * GPAvar) - log(theta / (1 - theta));
b1 = b1 * GPAvar / (GPAmu1 - GPAmu0);
b1
```

```
b1 = 2.9714
```

```
b2 = (GREmu1^2 - GREmu0^2) / (2 * GREvar) - log(theta / (1 - theta));
b2 = b2 * GREvar / (GREmu1 - GREmu0);
b2
```

```
b2 = 3.0971
```

```
miss = 0;
for i = 1:N
```

```
    if (GPA(i) > b1) ~= y(i)
        miss = miss + 1;
    end
end
GPAacc = 1 - miss / N;
GPAacc
```

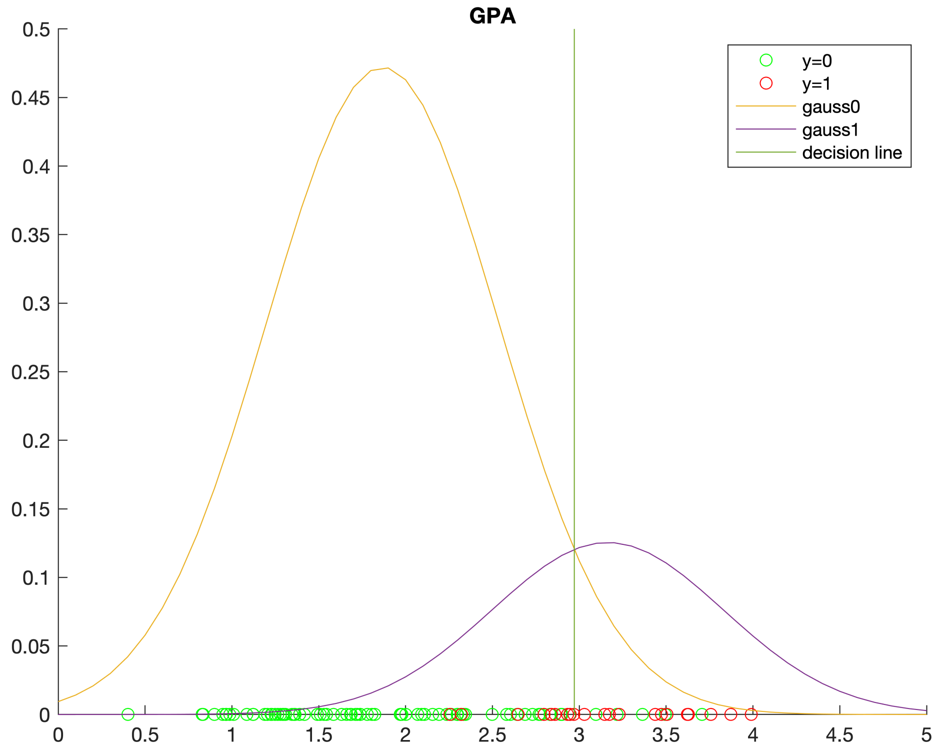
```
GPAacc = 0.8600
```

```
miss = 0;
for i = 1:N
    if (GRE(i) > b2) ~= y(i)
        miss = miss + 1;
    end
end
GREacc = 1 - miss / N;
GREacc
```

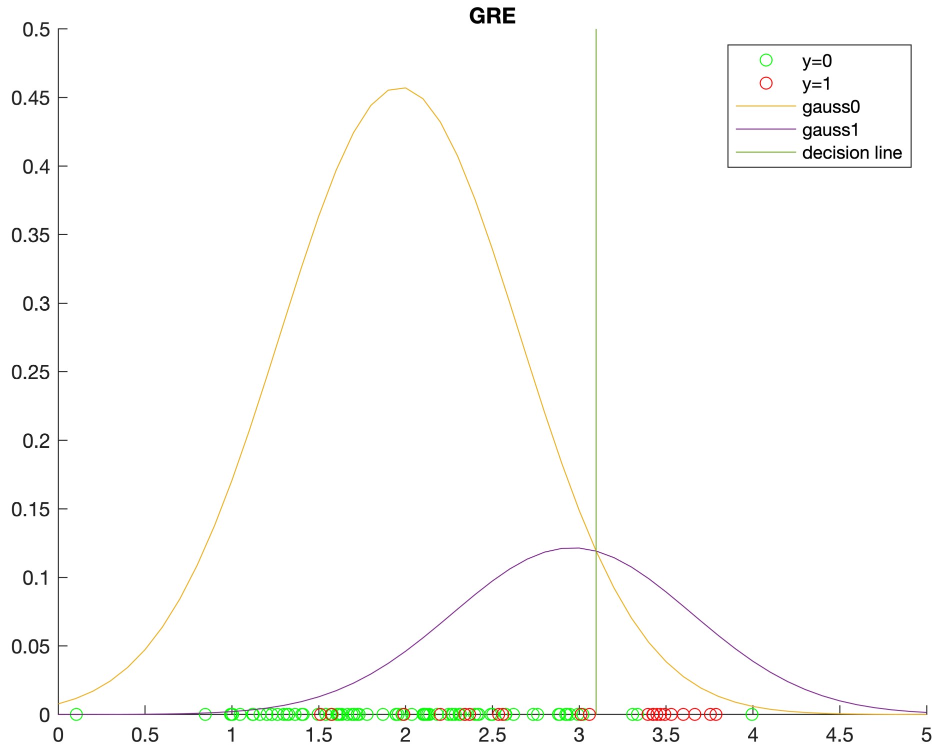
```
GREacc = 0.8700
```

c)

```
figure(1);
hold on;
scatter(GPA(y == 0), zeros(sum(y == 0), 1), 'green');
scatter(GPA(y == 1), zeros(sum(y == 1), 1), 'red');
gauss0 = Py0 * normpdf((0:0.1:5), GPAmu0, sqrt(GPAvar));
gauss1 = theta * normpdf((0:0.1:5), GPAmu1, sqrt(GPAvar));
plot((0:0.1:5), gauss0);
plot((0:0.1:5), gauss1);
plot(b1 * ones(51, 1), (0:0.01:0.5));
title("GPA");
legend('y=0', 'y=1', 'gauss0', 'gauss1', 'decision line');
```




```
figure(2);
hold on;
scatter(GRE(y == 0), zeros(sum(y == 0), 1), 'green');
scatter(GRE(y == 1), zeros(sum(y == 1), 1), 'red');
gauss0 = Py0 * normpdf((0:0.1:5), GREmu0, sqrt(GREvar));
gauss1 = theta * normpdf((0:0.1:5), GREmu1, sqrt(GREvar));
plot((0:0.1:5), gauss0);
plot((0:0.1:5), gauss1);
plot(b2 * ones(51, 1), (0:0.01:0.5));
title("GRE");
legend('y=0', 'y=1', 'gauss0', 'gauss1', 'decision line');
```



(d)

```
Py0
```

```
Py0 = 0.7900
```

```
GPAmu0
```

```
GPAmu0 = 1.8678
```

```
GPAmu1
```

```
GPAmu1 = 3.1637
```

```
GPAvar0 = 1 / (sum(y == 0)) * norm(GPA(y == 0) - GPAmu0)^2
```

```
GPAvar0 = 0.5066
```

```
GPAvar1 = 1 / (sum(y == 1)) * norm(GPA(y == 1) - GPAmu1)^2
```

```
GPAvar1 = 0.2163
```

```
GREmu0
```

```
GREmu0 = 1.9673
```

```
GREmu1
```

```
GREmu1 = 2.9590
```

```
GREvar0 = 1 / (sum(y == 0)) * norm(GRE(y == 0) - GREmu0)^2
```

```
GREvar0 = 0.4668
```

```
GREvar1 = 1 / (sum(y == 1)) * norm(GRE(y == 1) - GREmu1)^2
```

```
GREvar1 = 0.5035
```

(e)

```
gpaa = 1/2 * (1/GPAvar0 - 1/GPAvar1);
gpab = GPAmu1/GPAvar1 - GPAmu0/GPAvar0;
gpac = 1/2 * (GPAmu0^2/GPAvar0 - GPAmu1^2/GPAvar1) + log(theta / (1 - theta) * sqrt(GPA
gpar = roots([gpaa, gpab, gpac]);
gpar
```

```
gpar = 2×1
    5.3557
    2.9026
```

```
grea = 1/2 * (1/GREvar0 - 1/GREvar1);
greb = GREmu1/GREvar1 - GREmu0/GREvar0;
grec = 1/2 * (GREmu0^2/GREvar0 - GREmu1^2/GREvar1) + log(theta / (1 - theta) * sqrt(GRE
grer = roots([grea, greb, grec]);
grer
```

```
grer = 2×1
  -24.4141
    3.1038
```

```
miss = 0;
for i = 1:N
    x = GPA(i);
    if (x^2 * gpaa + x * gpab + gpac > 0) ~= y(i)
        miss = miss + 1;
    end
end
GPAacc = 1 - miss / N
```

```
GPAacc = 0.8800
```

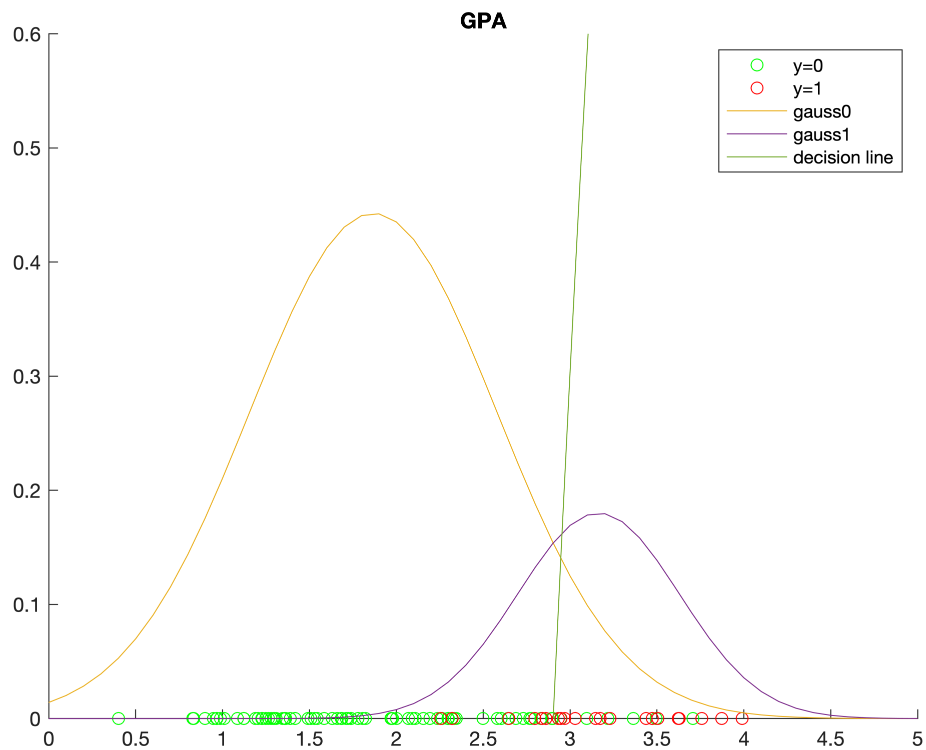
```
miss = 0;
for i = 1:N
    x = GRE(i);
    if (x^2 * grea + x * greb + grec > 0) ~= y(i)
        miss = miss + 1;
    end
end
GREacc = 1 - miss / N
```

```
GREacc = 0.8700
```

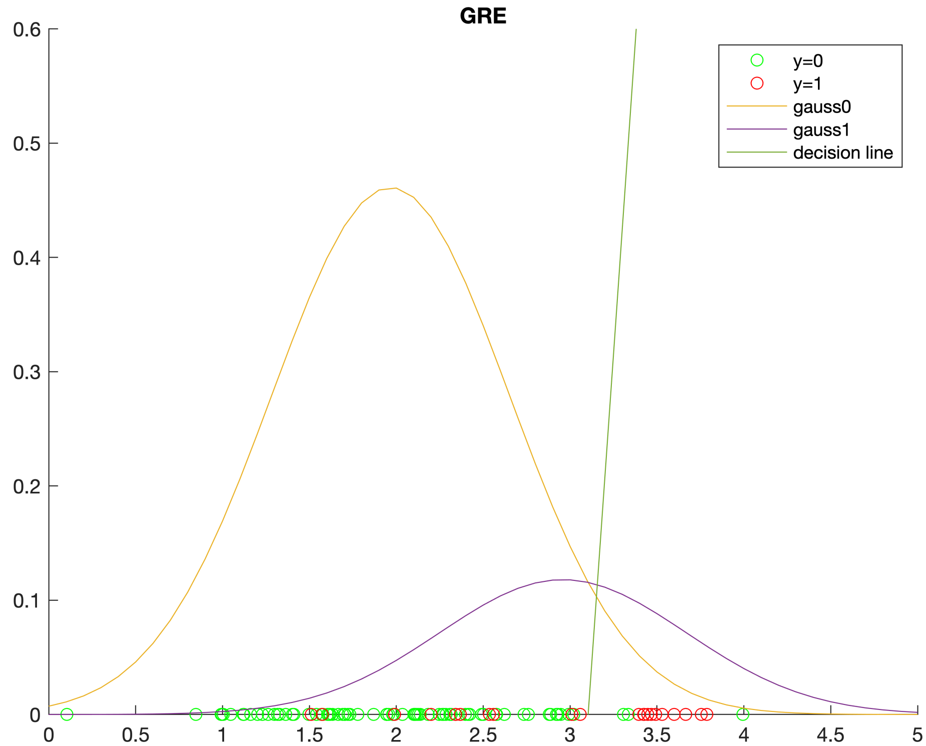
(f)

```
xpt = (0:0.01:5);
```

```
figure(3);
hold on;
scatter(GPA(y == 0), zeros(sum(y == 0), 1), 'green');
scatter(GPA(y == 1), zeros(sum(y == 1), 1), 'red');
gauss0 = Py0 * normpdf((0:0.1:5), GPAmu0, sqrt(GPAvar0));
gauss1 = theta * normpdf((0:0.1:5), GPAmu1, sqrt(GPAvar1));
plot((0:0.1:5), gauss0);
plot((0:0.1:5), gauss1);
ylim([0,0.6]);
plot(xpt, gpaa * xpt.^2 + gpab * xpt + gpac);
title("GPA");
legend('y=0', 'y=1', 'gauss0', 'gauss1', 'decision line');
```




```
figure(4);
hold on;
scatter(GRE(y == 0), zeros(sum(y == 0), 1), 'green');
scatter(GRE(y == 1), zeros(sum(y == 1), 1), 'red');
gauss0 = Py0 * normpdf((0:0.1:5), GREmu0, sqrt(GREvar0));
gauss1 = theta * normpdf((0:0.1:5), GREmu1, sqrt(GREvar1));
plot((0:0.1:5), gauss0);
plot((0:0.1:5), gauss1);
ylim([0,0.6]);
plot(xpt, grea * xpt.^2 + greb * xpt + grec);
title("GRE");
legend('y=0', 'y=1', 'gauss0', 'gauss1', 'decision line');
```


GRE
y=0
y=1
gauss0
gauss1
decision line
0.6
0.5
0.4
0.3
0.2
0.1
0
0
0.5
1
1.5
2
2.5
3
3.5
4
4.5
5