Machine Learning
Midsem

Q.1: $P(\text{Disease}) = \frac{1}{3000}$

$P(\text{No Disease}) = \frac{2999}{3000}$

$P(\text{test +ve} \mid \text{No disease}) = 0.005$

$P(\text{test -ve} \mid \text{disease}) = 0.025$

$P(\text{test-ve} \mid \text{No disease}) = 1 - 0.005 = 0.995$

$P(\text{test +ve} \mid \text{disease}) = 1 - 0.025 = 0.975$

Let x be the cost for each diagonistic test
Let n be the number of people tested.

→ Total cost of preventive treatment $= 125n\,[P(\text{test +ve} \mid \text{No disease})\,P(\text{No disease}) + P(\text{test +ve} \mid \text{disease})\,P(\text{disease})]$

$= 125 \times n \left[0.005 \times \left(\frac{2999}{3000}\right) + 0.975\left(\frac{1}{3000}\right)\right]$

$= 125 \times n\,[0.004998 + 0.000325]$

$= 0.665375n$

→ Total cost for people tested negative even though they have disease (have to take the chronical treatment) $= 25000\,[P(\text{test -ve} \mid \text{disease})\,P(\text{disease})] \times n$

$= 25000\left[0.025 \times \frac{1}{3000}\right] \times n$

$= 0.208333n$

→ Total diagonistic cost $= n \times x$

If ~~test~~ no test is taken, the people who have disease will have to the treatment.

So Total cost if no test taken $\Big\} = P(\text{disease}) \times n \times 25000$

$= \frac{1}{3000} \times 25000 \times n$

$= 8.3333n$

Now we know,

Diag cost + Total cost if no test taken $\leq$ T. cost of preventive treatment + T. cost of ppl tested negative though they have disease

~~$x \times n + 8.3333n \leq 0.20833n + 0.665375n$~~

~~$x \leq 8$~~

Total cost if not test taken $\geq$ Diagnostic cost + T. cost of preventive treatment + T. cost of ppl tested -ve though they have disease

$8.3333n \geq x n + 0.20833n + 0.665375n$

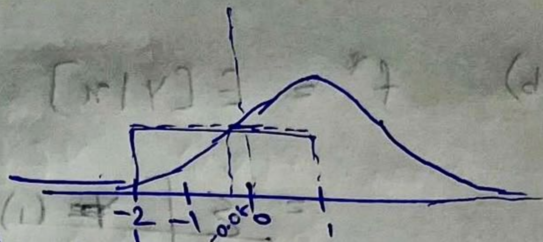
$x \leq 8.3333 - 0.873705$

$\boxed{x \leq 7.46 \text{ dollars}}$

Therefore the diagnostic test must be less than 7.46 dollars

Q.2 · a) $P(Y=1) = 0.4$

$P(Y=-1) = 0.6$



$P(x \mid Y=1) = \frac{1}{b-a} = \frac{1}{+2+(+1)} = \frac{1}{3} = f_1$

$= 1-(-2)$

$P(x \mid Y=-1) = N(1,1) = \frac{1}{\sqrt{2\pi}} e^{-\frac{(x-1)^2}{2}} = f_2$

$$P(Y=1 \mid x) = \frac{P(x \mid Y_{=1}) P(Y_{=1})}{P(x)}$$

$$= \frac{\frac{1}{3} \times 0.4}{\frac{0.4}{3} + 0.6\left(\frac{1}{\sqrt{2\pi}} e^{-\frac{(x-1)^2}{2}}\right)}$$

$$P(Y=-1 \mid x) = \frac{P(x \mid Y=-1)\ P(Y=-1)}{P(x)}$$

$$= \frac{\frac{0.6}{\sqrt{2\pi}} e^{-\frac{(x-1)^2}{2}}}{\frac{0.4}{3} + \frac{0.6}{\sqrt{2\pi}} e^{-\frac{(x-1)^2}{2}}}$$

Bayes optimal Predictor:

$$f^* = \underset{y}{\operatorname{argmax}} \left(P(y=1 \mid x),\ P(y=-1 \mid x) \right)$$

$$= \underset{y}{\operatorname{argmax}} \left(\frac{0.4}{0.4 + \frac{2.4}{\sqrt{2\pi}} e^{-\frac{(x-1)^2}{2}}},\ \frac{\frac{2.4}{\sqrt{2\pi}} e^{-\frac{(x-1)^2}{2}}}{0.4 + \frac{2.4}{\sqrt{2\pi}} e^{-\frac{(x-1)^2}{2}}} \right)$$

From ~~2b~~ 2c, we found classification point is -0.0818.

So,

$$\text{Bayes optimal predictor} = \begin{cases} +1 & -2 \le x \le -0.081 \\ -1 & \text{otherwise} \end{cases}$$

Note: [steps for this classification point is in 2c], I wrote that and finished this question later.

2.b) $f^* = E[Y|x]$

$$= \cancel{E\left[(1)\,P(y=1|x) + (-1)\,P(y=-1|x)\right]}$$

$$= \sum_y y\, f(y|x)$$

$$= (1)P(y=1|x) + (-1)\,P(y=-1|x)$$

$$= P(y=1|x) - P(y=-1|x)$$

Interval $(-\infty, -2)$

$$E[Y|x] = \frac{0 - \frac{0.6}{\sqrt{2\pi}} e^{-(x-1)^2/2}}{P(x)}$$

$$= \frac{-\frac{0.6}{\sqrt{2\pi}} e^{-(x-1)^2/2}}{\frac{0.4}{3} + \frac{0.6}{\sqrt{2\pi}} e^{-(x-1)^2/2}}$$

Interval $[-2, 1]$

$$E(Y|x) = \frac{\frac{0.4}{3} - \frac{0.6}{\sqrt{2\pi}} e^{-(x-1)^2/2}}{\frac{0.4}{3} + \frac{0.6}{\sqrt{2\pi}} e^{-(x-1)^2/2}}$$

Interval $(1, \infty)$

$$E[Y|x] = \frac{-\frac{0.6}{\sqrt{2\pi}} e^{-(x-1)^2/2}}{\frac{0.4}{3} + \frac{0.6}{\sqrt{2\pi}} e^{-(x-1)^2/2}}$$

$$E[Y|x] = \begin{cases} \dfrac{-\frac{0.6}{\sqrt{2\pi}} e^{-(x-1)^2/2}}{\frac{0.4}{3} + \frac{0.6}{\sqrt{2\pi}} e^{-(x-1)^2/2}} & ;\ \text{for } x \in [-\infty, -2) \cup (1, \infty) \\[2ex] \dfrac{\frac{0.4}{3} - \frac{0.6}{\sqrt{2\pi}} e^{-(x-1)^2/2}}{\frac{0.4}{3} + \frac{0.6}{\sqrt{2\pi}} e^{-(x-1)^2/2}} & ;\ \text{for } x \in [-2, 1] \end{cases}$$

2.c) Expected Loss $= \underset{x,y}{E} \ \underset{y}{\operatorname{argmin}} \left(P(y=1|x),\ P(y=-1|x) \right)$

To find Classification point for $y=+1$ & $y=-1$, equate & find x, using:

$$P(y=1|x) = P(y=-1|x)$$

$$\frac{\frac{\overset{2}{\cancel{0.4}}}{3}}{\cancel{P(x)}} = \frac{\frac{\overset{3}{0.6}}{\sqrt{2\pi}} e^{-\frac{(x-1)^2}{2}}}{\cancel{P(x)}}$$

$$e^{-\frac{(x-1)^2}{2}} = \frac{2\sqrt{2\pi}}{9}$$

$$-\frac{(x-1)^2}{2} = \ln\left(\frac{2\sqrt{2\pi}}{9}\right)$$

$$(x-1)^2 = -2\ln\left(\frac{2\sqrt{2\pi}}{9}\right)$$

$$\Rightarrow \quad x-1 = \sqrt{-2\ln\left(\frac{2\sqrt{2\pi}}{9}\right)} \ (\text{or}) \ -x+1 = -2\ln\left(\frac{2\sqrt{2\pi}}{9}\right)$$

$$x = 1 + 1.0818 \ (\text{or}) \ x = 1 - 1.0818$$

$$x = 2.0818 \ (\text{or}) \ \boxed{x = -0.0818}$$

x can't be 2.0818 as the limits of x in f_1 is $[-2,1]$ and can't overlap at 2.0818.

$$\text{Expected loss} = \int_{-\infty}^{-2} 0\,dx + \int_{-2}^{-0.0818} \frac{\frac{0.6}{\sqrt{2\pi}} e^{-\frac{(x-1)^2}{2}}}{P(x)}\,dx + \int_{-0.0818}^{1} \frac{\frac{0.4}{3}}{P(x)}\,dx + \int_{1}^{\infty} 0\,dx$$

$$= 0 + 0.083 + \frac{0.4}{3} x \Big]_{-0.0818}^{1} + 0$$

$$= 0.083 + \cancel{0.2424}\ 0.14424$$

$$\cancel{= 0.3254}$$

$$= 0.22724$$

MidTerm_ML.m × | HW1.m × | MT_2.m × | MT-ML.m × | try_2.m × | Trymt.m

```matlab
syms x;
f1=@(x)(0.6*normpdf(x,1,1));
f2=@(x)(0.4*unifpdf(x,-2,1));
a1=int(f2(x),x,-2,-0.0818)

q=0.4
f3=@(x)((1-(2/3)*0.6*normpdf(x,1,1)-q/2))

q4_1=integral(f2,-0.0818,1)+integral(f1,-2,-0.0818)

```

Command Window

New to MATLAB? See resources for Getting Started.

```
q4_1 =

    0.2272
```

P.3. Given: $P(x,y)$, $y \in \{-1,1\}$

$P_q(x) = P(x)$

$P_q(y|x) = (1-q)\,P(y|x) + q\,u(-1,1)$

$P(u=-1) = P(u=1) = \frac{1}{2}$

$$x_i \begin{cases} \text{wp } q \rightarrow y_i \text{ is random} \\ \text{wp } (1-q) \rightarrow y_i \text{ is from } P(y|x_i) \end{cases}$$

① Bayes optimal predictor: $P(x,y)$

$$y^* = \begin{cases} +1 & P(y=1|x) > 0.5 \\ -1 & P(y=-1|x) > 0.5 \end{cases}$$

Assuming $P_q(y=1|x) > 0.5$

$\Rightarrow P_q(y=1|x) > P_q(y=-1|x)$

$\cancel{(1-q)}\,P(y=1|x) + \cancel{q\,u(-1,1)} > \cancel{(1-q)}\,P(y=-1|x) + \cancel{q\,u(-1,1)}$

$P(y=1|x) > P(y=-1|x)$

$P(y=1|x) > 1 - P(y=1|x)$

$2P(y=1|x) > 1$

$P(y=1|x) > \frac{1}{2}$

Bayes optimal predictor for y^*_q:

$$y^*_q = \begin{cases} +1 & P(y=1|x) > 0.5 \\ -1 & P(y=-1|x) > 0.5 \end{cases}$$

$\therefore$ Both bayes opt. predictor for y^* & y^*_q are same.

3 2.

Expected loss of Bayes opt. predictor : $P(x,y)$

$$R^* = E[\min(P(y=1|x), P(y=-1|x))]$$

$$= E[\min(P(y=1|x), 1-P(y=1|x))]$$

Expected loss of Bayes opt. predictor: $P_q(x,y)$

$$R_q^* = E[\min(P_q(y=1|x), P_q(y=-1|x))]$$

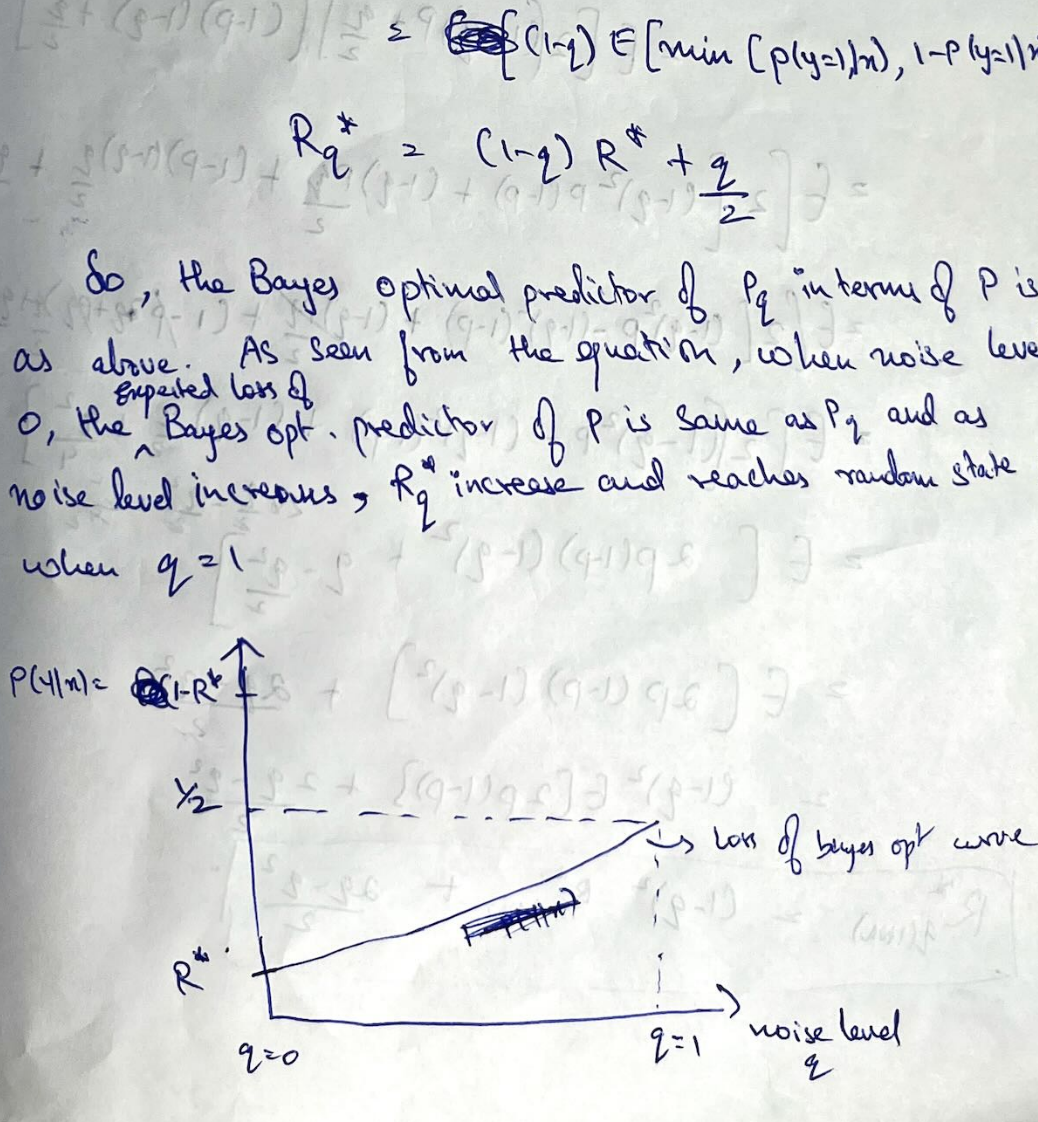
$$= E\left[\min\left(\left((1-q)P(y=1|x)+\frac{q}{2}\right), \left((1-q)(1-P(y=1|x))+\frac{q}{2}\right)\right)\right]$$

$$= E\left[\min[((1-q)P(y=1|x)), (1-q)(1-P(y=1|x))] + \frac{q}{2}\right]$$

$$= (1-q)\, E[\min[P(y=1|x), 1-P(y=1|x)]] + \frac{q}{2}$$

$$R_q^* = (1-q)R^* + \frac{q}{2}$$

So, the Bayes optimal predictor of P_q in terms of P is as above. As seen from the equation, when noise level is 0, the expected loss of Bayes opt. predictor of P is same as P_q and as noise level increases, R_q^* increase and reaches random state $\frac{1}{2}$ when $q=1$



3.3. Expected classification loss of 1NN : $P(x,y)$

Assuming $P(y=1|x) > P(y=-1|x)$,

$$R^*_{1\text{-}NN} = E\left[\, 2\,p(y=1|x)\,(1-p(y=1|x))\right]$$

Let $p(y=1|x) = p$, $P_q(y=1|x) = P_q$

$$\boxed{R^*_{1\text{-}NN} = E[2p(1-p)]}$$

Expected classification loss of 1NN : $P_q(x,y)$

$$R^*_{q(1\text{-}NN)} = E[2P_q(1-P_q)]$$

$$= E\left[2\left((1-q)p + \frac{q}{2}\right)\left[1-(1-q)p - \frac{q}{2}\right]\right]$$

$$= E\left[2\left(\overset{p-qp+q/2}{(1-q)p+\frac{q}{2}}\right)\left(\left(1-p+pq-\frac{q}{2}\right)+q-\frac{q}{2}\right)\right]$$

$$= E\left[2\left((1-q)p+\frac{q}{2}\right)\left((1-p)(1-q)+\frac{q}{2}\right)\right]$$

$$= E\left[2\left[(1-q)^2p(1-p) + (1-q)\frac{pq}{2} + (1-p)(1-q)\frac{q}{2} + \frac{q^2}{4}\right]\right]$$

$$= E\left[2\left[(1-q)^2p - (1-q)^2(1-p) + \cancel{(1-q)\frac{pq}{2}} + (1-\cancel{p}-q+\cancel{pq})\frac{q}{2} + \frac{q^2}{4}\right]\right]$$

$$= E\left[2\left[(1-q)^2p - (1-q)^2(1-p) + (1-q)\frac{q}{2} + \frac{q^2}{4}\right]\right]$$

$$2\times\frac{q}{2} - \frac{q^2}{2} + \frac{q^2}{4}2$$

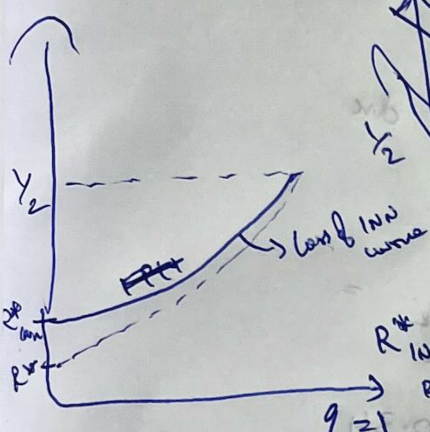
$$= E\left[\, 2p(1-p)(1-q)^2 + q - \frac{q^2}{2}\right]$$

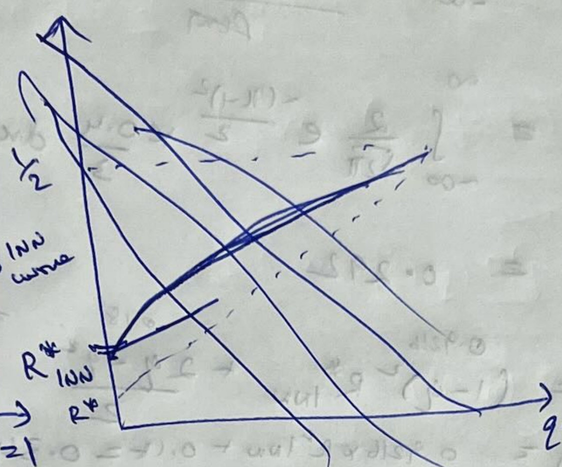
$$= E\left[2p(1-p)(1-q)^2\right] + \frac{2q-q^2}{2}$$

$$= (1-q)^2\, E[2p(1-p)] + \frac{2q-q^2}{2}$$

$$\boxed{R^*_{q(1NN)} = (1-q)^2\, R^*_{1NN} + \frac{2q-q^2}{2}}$$

As seen above, the expected loss of $R^*_{q(1\text{-}NN)}$ is equal to R^*_{1NN} when noise level $q = 0$, and increases to $1/2$ as the noise level goes to 1. Since this is a quadratic function, the curve is parabolic.



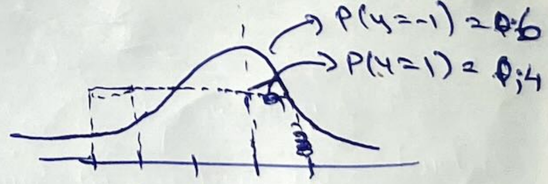


Note :

$R^*_N < \frac{R^*_{1NN}}{2}$

3.4)

i) $P(y|x) = \begin{cases} \dfrac{\frac{0.6}{\sqrt{2\pi}} e^{-\frac{(x-1)^2}{2}}}{P(x)} & \text{if } y=-1 \\ \dfrac{\frac{0.4}{3}}{p(x)} & \text{if } y = 1 \end{cases}$



$R^* = E\left[\min\left(P(y=1|x), 1-P(y=1|x)\right)\right]$

$= \int_{-\infty}^{-2} 0\,dx + \int_{-2}^{-0.0818} P(x)\times\frac{0.6}{\sqrt{2\pi}}\frac{e^{-\frac{(x-1)^2}{2}}}{P(x)}\,dx + \int_{-0.0818}^{1}\frac{0.4}{3P(x)}\times P(x)\,dx + \int_{1}^{\infty} 0\,dx$

$= \cancel{0.0272}\ 0.2272$

suppose if $q = 0.2$

$R^*_q = (1-q)R^* + \frac{q}{2}$

$= 0.8 \times 0.2272 + 0.1$

$= 0.28176$

If $q = 0$

$R^*q = 1 \times R^* = 0.2272$

If $q = 1$

$R^*_q = 0 \times R^* + \frac{1}{2} = 0.5$

As seen from the code below, the value are almost similar to the original theoretical values. Hence ~~proved~~ verified empirically

MidTerm_ML.m | HW1.m | MT_2.m | MT-ML.m | try_2.m | Trymt.m | trytry.m | Midterm_2_finaltry.m | untitled3 *

```matlab
a= [0,0.2,1]; % Set up parameters
for i = 1:length(a)
    q = a(i);
    num_samples = 10000*(1-q); % number of samples to generate
    noise_samples = 10000*q;
    p_y1 = 0.4; % probability of Y=1
    p_y2 = 0.6; % probability of Y=-1
    y1_range = [-2, 1]; % range of uniform distribution for Y=1
    y2_mean = 1; % mean of normal distribution for Y=-1
    y2_std = 1; % standard deviation of normal distribution for Y=-1
    % Generate samples
    ys = randsample([-1, 1], num_samples, true, [p_y2, p_y1]); % randomly choose Y for each sample
    xs = zeros(num_samples, 1); % initialize X
    for i = 1:num_samples
        if ys(i) == 1
            xs(i) = unifrnd(y1_range(1), y1_range(2)); % choose X from uniform distribution
        else
            xs(i) = normrnd(y2_mean, y2_std); % choose X from normal distribution
        end
    end
    yq = randsample([-1, 1], noise_samples, true, [0.5,0.5]); % randomly choose Y for each sample
    xq = zeros(noise_samples, 1)-1000000; % initialize X, considering that the chance of  x=-1000000 is always negligible, and the result is always random
    y = horzcat(yq, ys); x = vertcat(xq, xs);
    count_loss=0;
    for i =1:10000
        if y(i)==1 && x(i)<(-2)
            count_loss=count_loss+1;
        elseif y(i)==-1 && x(i)<-0.0818 && x(i)>-2
            count_loss=count_loss+1;
        elseif y(i)==1 && x(i)>-0.0818
            count_loss=count_loss+1;
        end
    end
    rstarq=count_loss/10000
end
```

Command Window

New to MATLAB? See resources for Getting Started.

```
    0.2329


rstarq =

    0.2845


rstarq =

    0.5033
```

ii) $R^*_{1NN} = E[2p(1-p)]$

$$= E[2P(y=1|x)\,P(y=-1|x)]$$

$$= \int_{-\infty}^{\infty} 2\,\frac{\frac{e^{-\frac{(x-1)^2}{2}}}{\sqrt{2\pi}} \times \frac{0.4}{3} \times \cancel{P(x)}}{\cancel{P(x)}}\,dx$$

$$= \int_{-\infty}^{\infty} \frac{2}{\sqrt{2\pi}} e^{-\frac{(x-1)^2}{2}} \times \frac{0.4}{3}\,dx$$

$$= 0.292$$

$$R^*_{q(1NN)} = \overset{0.9216}{(1-q)^2}\, R^*_{1NN} + \overset{0.18}{\frac{2q-q^2}{2}} =$$

$q=0.2 \Rightarrow R^*_{q(1NN)} = 0.9216 \times R^*_{1NN} + 0.18 = 0.34$

$q=1 \Rightarrow R^*_{q(1NN)} = 0 + \frac{2-1}{2} = \frac{1}{2} = 0.5$

As seen in the code below, the value of noise KNN is almost similar to the ones predicted

MidTerm_ML.m | HW1.m | MT_2.m | MT-ML.m | try_2.m | Trymt.m | trytry.m | Midterm_2_finaltry.m | MT_fin.m

```matlab
a= [0,0.2,1]; % Set up parameters
for i = 1:length(a)
    q = a(i);
    num_samples = 10000*(1-q); % number of samples to generate
    noise_samples = 10000*q;
    p_y1 = 0.4; % probability of Y=1
    p_y2 = 0.6; % probability of Y=-1
    y1_range = [-2, 1]; % range of uniform distribution for Y=1
    y2_mean = 1; % mean of normal distribution for Y=-1
    y2_std = 1; % standard deviation of normal distribution for Y=-1
    % Generate samples
    ys = randsample([-1, 1], num_samples, true, [p_y2, p_y1]); % randomly choose Y for each sample
    xs = zeros(num_samples, 1); % initialize X
    for i = 1:num_samples
        if ys(i) == 1
            xs(i) = unifrnd(y1_range(1), y1_range(2)); % choose X from uniform distribution
        else
            xs(i) = normrnd(y2_mean, y2_std); % choose X from normal distribution
        end
    end
    yq = randsample([-1, 1], noise_samples, true, [0.5,0.5]); % randomly choose Y for each sample
    xq = zeros(noise_samples, 1)-1000000; % initialize X, considering that the chance of  x=-1000000 is always negligible, and the result is always random
    y = horzcat(yq, ys); x = vertcat(xq, xs);
    y = reshape(y, [], 1);
    cv = cvpartition(size(x,1),'HoldOut',0.3);
    idx = cv.test;
    Xtrain = x(~idx,:); Ytrain = y(~idx,:); Xtest  = x(idx,:); Ytest  = y(idx,:);
    mdl = fitcknn(Xtrain,Ytrain,'NumNeighbors',1); % Create KNN model
    Ypred = predict(mdl,Xtest);% Make predictions using the KNN model
    Loss = 1- sum(Ypred == Ytest) / numel(Ytest);
    Loss
end
```

Command Window

New to MATLAB? See resources for Getting Started.

```
Loss =

    0.2603


Loss =

    0.3417


Loss =

    0.5150
```