Problem 1

Please find the attached code:

1. setLabel - this returns the value {-1, 1} depending on the value of y
2. returnRisk - this returns the average risk using the empirical risk formula
3. returnError - this return the average error
4. LearningRate - the weights, iterations, error_list, risk_list
    a. risk_list - calculated using the above described function returnRisk
    b. error-list - calculated using the above described function returnError
    c. iterations - the number of times the if condition is satisfied
    d. weights - this is calculated using the eetah, x and y
5. show_model - code to display the model graph(decision boundary)
6. Show_err_graph - code to display the error graph(error/risk vs Iterations)

Formulas used:

Empirical Risk,

$$R(\boldsymbol{\theta}) = \frac{1}{N} \sum_{i=1 to N} step(-y_i \boldsymbol{\theta}^T x_i)$$

Binary Error,

$$\mathrm{E}(\boldsymbol{\theta}) = \frac{N_{misclassified}}{N_{total}}$$

For SGD,

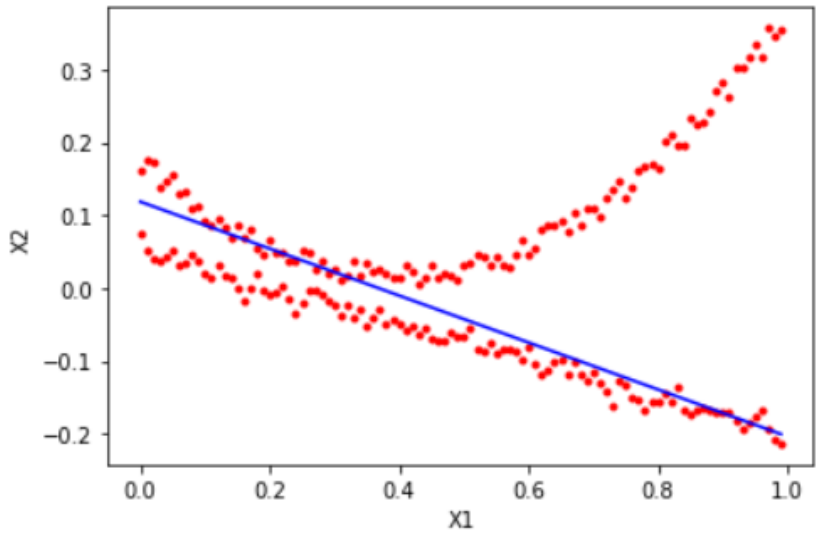
$$\boldsymbol{\theta}^{t+1} = \boldsymbol{\theta}^t - \eta \nabla_{\boldsymbol{\theta}} R^{per}|_{\boldsymbol{\theta}_t} = \boldsymbol{\theta}^t + y_i x_i \quad \text{-------- } \eta=1$$

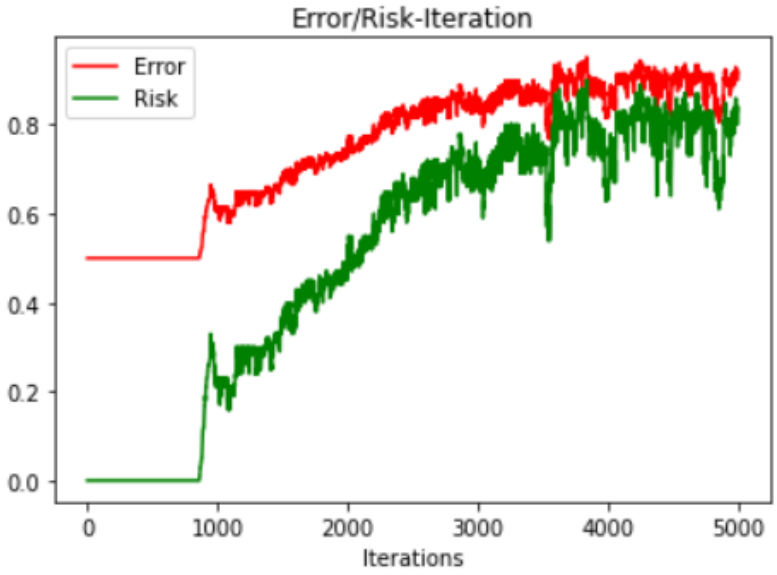
While training, the weight has been initialized randomly.
Max iteration was set to 5000

The plot of the Decision boundary:

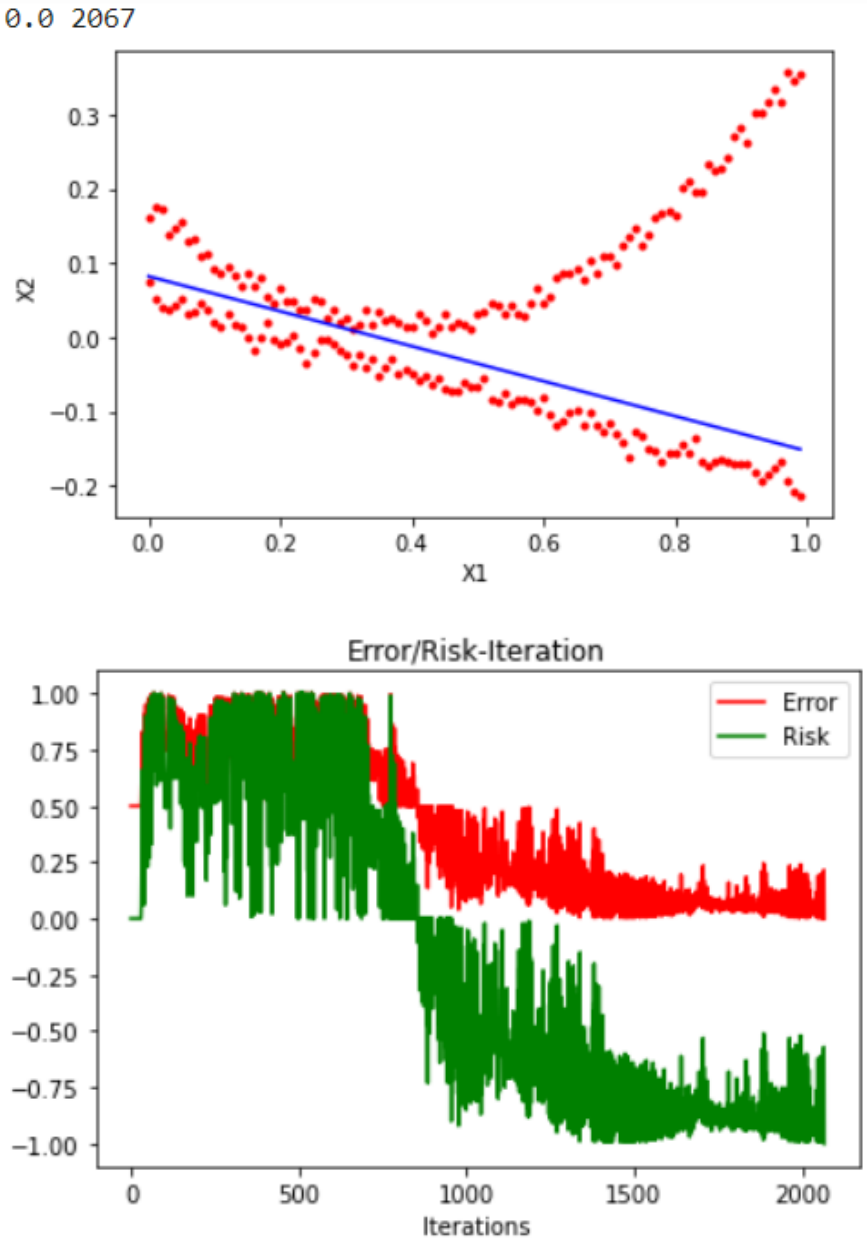
Learning rate = 0.001
Iterations taken to converge to 0 = 4999

0.92 4999



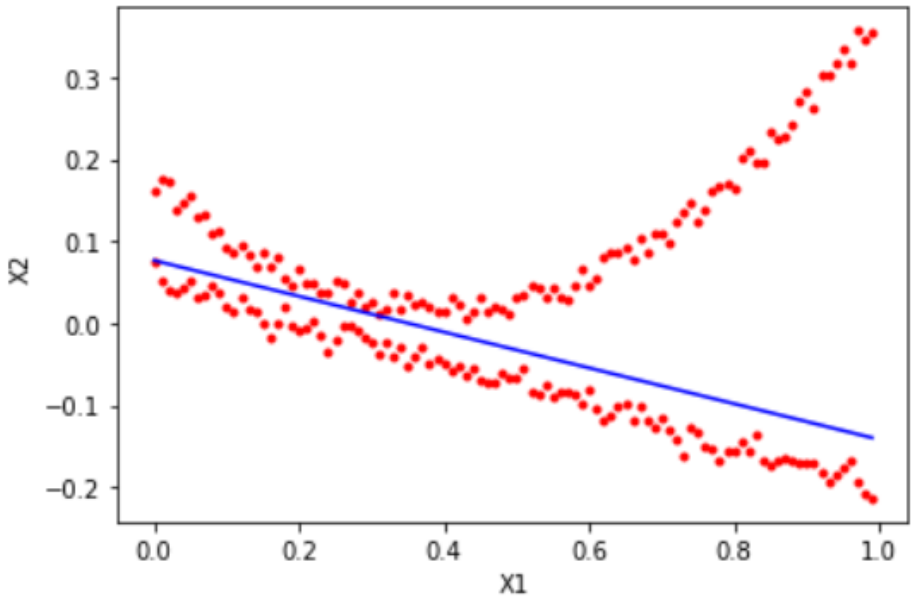


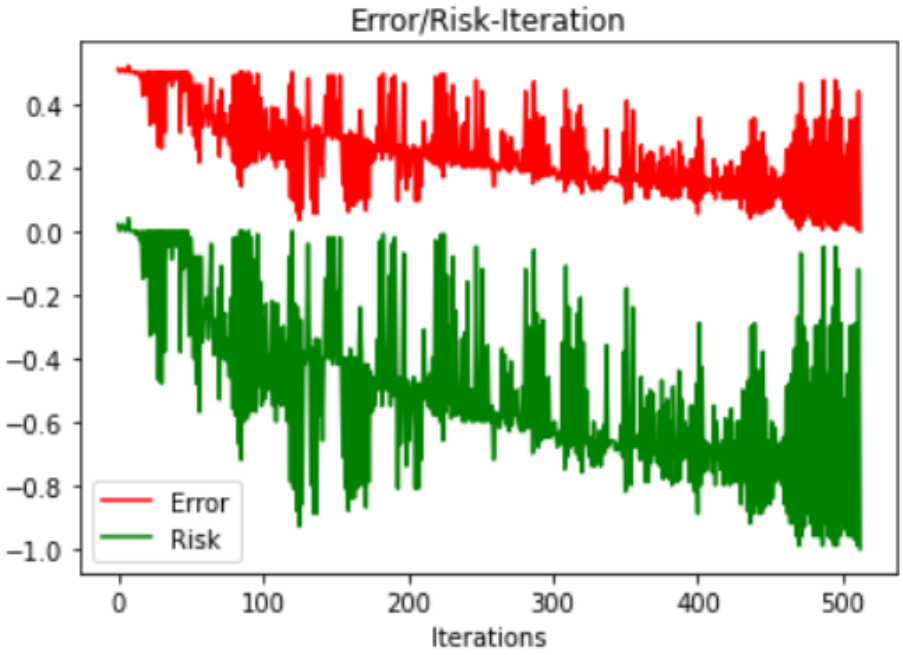
Step size: 0.01
Iterations to converge to 0: 2067



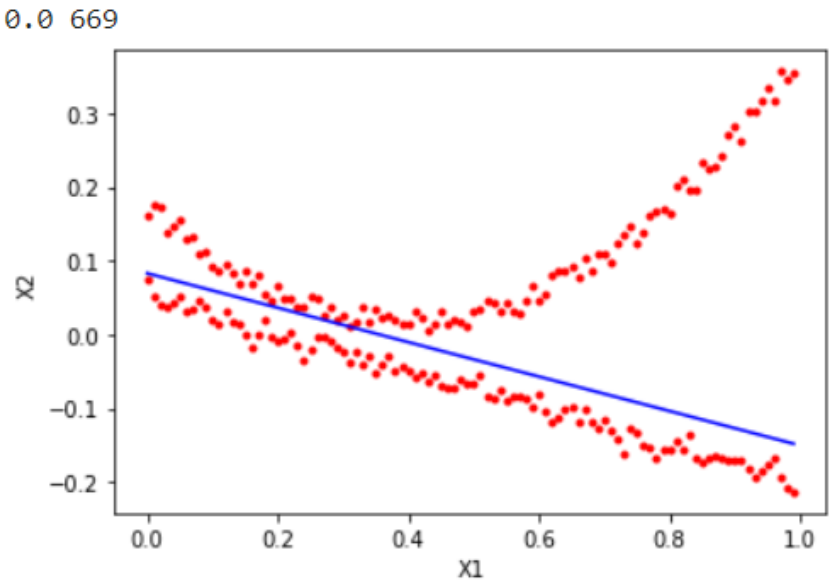
Step size: 0.02
Iterations to converge to 0: 512

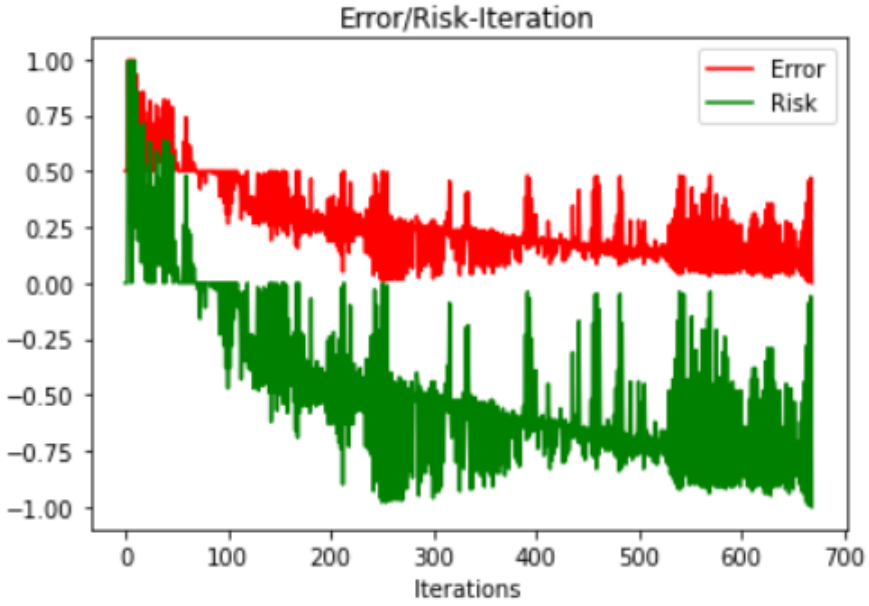
```
0.0 512
```





Step size: 0.1
Iterations to converge to 0: 669





As per the observations by changing the value of learning rate as the learning rate increases the number of iterations decreases.

Problem 2. a. :

Loss function : $E = -\sum_i (t_i \log(x_i) + (1 - t_i) \log(1 - x_i))$

$$x_i = \frac{1}{1+e^{-s_i}}$$

$$y_j = \frac{1}{1+e^{-s_j}}$$

$where\ s_i = \sum_j y_j w_{ji}$

Perform back propagation,

$$\frac{\partial E}{\partial w_{ji}} = \frac{\partial E}{\partial x_i}\frac{\partial x_i}{\partial s_i}\frac{\partial s_i}{\partial w_{ji}}$$

$$\frac{\partial E}{\partial x_i} = -\frac{t_i}{x_i} + \frac{1-t_i}{1-x_i} = \frac{x_i - t_i}{x_i(1-x_i)}$$

$$\frac{\partial x}{\partial s_i} = x_i(1-x_i)$$

$$\frac{\partial s}{\partial w_{ji}} = y_j$$

$$\frac{\partial E}{\partial w_{ji}} = y_j(x_i - t_i)$$

Assume $(x_i - t_i) = \delta_i = \frac{\partial E}{\partial x_i}\frac{\partial x_i}{\partial s_i} = \frac{\partial E}{\partial s_i}$ ----------1

These equations give us the gradient of the errors in the last layer of the network. Now we need to calculate the gradient of the error with respect to the lower layer of the network(i.e. Connecting the inputs of the hidden layer units). So we apply the same chain rule. This is known as the backpropagation algorithm.

Denote the weight between the hidden layer and the input layer as w_{kj},perform backpropagation on w_{kj}

$s_i = \sum_j y_j w_{ji}\ similarly\ s_j = \sum_i x_i w_{kj}$

$x_i = \frac{1}{1+e^{-s}}$

Now we have s_j in terms of w_{kj}

E in terms of x_i

$$\frac{\partial E}{\partial w_{kj}} = \frac{\partial E}{\partial s_j}\frac{\partial s_j}{\partial w_{kj}}$$

$$= \sum_i \frac{\partial E}{\partial s_i}\frac{\partial s_i}{\partial s_j}\frac{\partial s_j}{\partial w_{kj}}$$

Substitute $\frac{\partial E}{\partial s_i}$ from equation 1

$$= \sum_i \delta_i \frac{\partial s_i}{\partial s_j} \frac{\partial}{\partial w_{kj}} \sum_j w_{kj} z_k$$

$$\frac{\partial E}{\partial w_{kj}} = \sum_i \delta_i z_k \frac{\partial s_i}{\partial s_j} \text{ ----------2}$$

Now we should calculate $\frac{\partial s_i}{\partial s_j}$

$$\frac{\partial s_i}{\partial s_j} = \frac{\partial s_i}{\partial y_j} \frac{\partial y_j}{\partial s_j}$$

$$= \frac{\partial}{\partial y_j} \sum_j y_j w_{ji} \frac{\partial y_j}{\partial s_j} \quad \text{----------> } \left(y_j = \frac{1}{1+e^{-s_j}}\right)$$

$$\frac{\partial s_i}{\partial s_j} = \sum_j w_{ji} y_j (1 - y_j)$$

Substitute $\frac{\partial s_i}{\partial s_j}$ in equation 2

$$\frac{\partial E}{\partial w_{kj}} = \sum_i \delta_i z_k \sum_j w_{ji} y_j (1 - y_j)$$

$$= z_k \delta_i w_{ji} y_j (1 - y_j)$$

Assume $\delta_j = \delta_i w_{ji} y_j (1 - y_j)$

$$\frac{\partial E}{\partial w_{kj}} = z_k \delta_j = (x_i - t_i) w_{ji} y_j (1 - y_j)$$

2. b.

Loss function: $E = -\sum_i (t_i \log(x_i))$

Softmax activation function: $x_i = \frac{e^{s_i}}{\sum_{c=1 \text{ to } m} e^{s_c}}$

$where\ s_i = \sum_j y_j w_{ji}$

Perform back propagation,

$$\frac{\partial E}{\partial w_{ji}} = \frac{\partial E}{\partial x_i} \frac{\partial x_i}{\partial s_i} \frac{\partial s_i}{\partial w_{ji}}$$

$$\frac{\partial E}{\partial x_i} = -\frac{t_i}{x_i}$$

$$\frac{\partial x}{\partial s_i} = \frac{\partial x}{\partial s_i}\frac{e^{si}}{\sum_i e^{si}} = \frac{(e^{si})'\sum_i e^{si} - e^{si}(\sum_i e^{si})'}{(\sum_i e^{si})^2} = x_i - x_i^2 = x_i(1 - x_i)$$

$$\frac{\partial s}{\partial w_{ji}} = y_j$$

Substituting above 3 in $\frac{\partial E}{\partial w_{ji}}$

$$\frac{\partial E}{\partial w_{ji}} = -\frac{t_i}{x_i} x_i(1 - x_i) y_j = -t_i(1 - x_i) y_j$$

Assume $-t_i(1 - x_i) = \delta_i$

Now perform back propagation for w_{kj}:

$$\frac{\partial E}{\partial w_{kj}} = \sum_i \frac{\partial E}{\partial s_j}\frac{\partial s_j}{\partial w_{kj}}$$

$$= \sum_i \frac{\partial E}{\partial s_j}\frac{\partial}{\partial w_{kj}} w_{kj} z_k$$

$$= \sum_i \frac{\partial E}{\partial s_i}\frac{\partial s_i}{\partial s_j} z_k$$ ----------4

We need to calculate $\frac{\partial E}{\partial s_i}\frac{\partial s_i}{\partial s_j}$

$$\frac{\partial E}{\partial s_i}\frac{\partial s_i}{\partial s_j} = \frac{\partial E}{\partial x_i}\frac{\partial x_i}{\partial s_i}\frac{\partial s_i}{\partial y_j}\frac{\partial y_j}{\partial s_j}$$

$$= \sum_i \delta_i w_{ji} \frac{\partial}{\partial s_j}\frac{e^{s_i}}{\sum_{c=1\ to\ m} e^{s_c}}$$

$$= \sum_i \delta_i y_j(y_j + 1)$$

Substitute in 4

$$\frac{\partial E}{\partial w_{kj}} = \sum_i \delta_i y_j(y_j + 1) z_k$$

Problem 3:

We need to maximize the entropy using lagrange multiplier
Given function:

H = $\sum_{k=1toN} p_k log p_k$

Consider distribution $\{p_k | k = 1,2,...,N\}$ is a N-dimensional vector x which meets the constraint of $\sum_{i=1toD} x_i = 1$

Hence H = $x^T log x$

Rewrite x in terms of I(Identity vector). Since we need to use equality = 0 for lagrange multipliers

$I^T x = 1$

i.e. $I^T x$ -1 =0

Using Lagrange multiplier λ to optimize under the above constrain,

$H_{max} = H' - \lambda(equality = 0)$

$= H - \lambda(I^T x - 1)$

$H' = \frac{\partial}{\partial x}(x^T log x - \lambda(I^T x - 1))$

$= I + log x - \lambda I$

Set H' = 0 to get the value of x,

$I + log x - \lambda I = 0$

logx = $(\lambda - 1) I$

$x = e^{\lambda - 1} I$ ----------1

substitute $I^T x = 1$

$I^T e^{\lambda - 1} I = 1$

$e^{\lambda - 1}$N = 1

$\lambda = 1 - log N$

Re substitute the value of λ in equation 1

$x = e^{1 - log N - 1} I$

$$= \frac{1}{N} I$$

Which means, the distribution{pk|k = 1,2,...,N}to maximize the entropy is {pk = $\frac{1}{N}$|k=1,2,3...N}

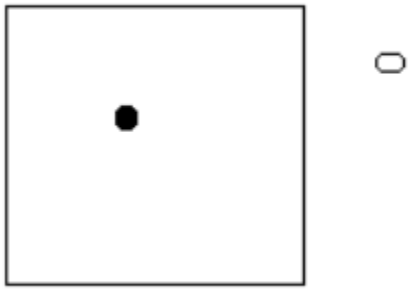
Problem 4:

VC Dimension H is defined as: the maximum number of points h that can be arranged so that f(x) can shatter them.

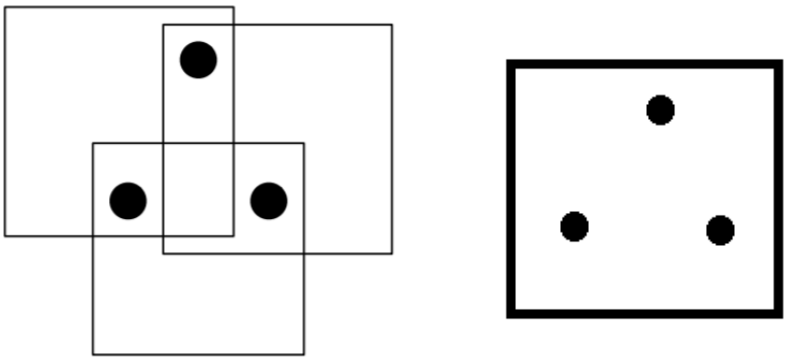
For axis aligned square the **VC dimension is 3**

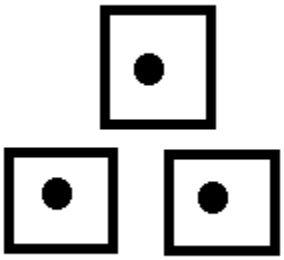
To prove this

1. Consider 1 point that can be shattered.
   This can be shattered by an axis aligned square since it will either be considered inside or outside the square.
2. Consider 2 points. Here again we can shatter 2 points easily since one will be inside and the other can be outside.



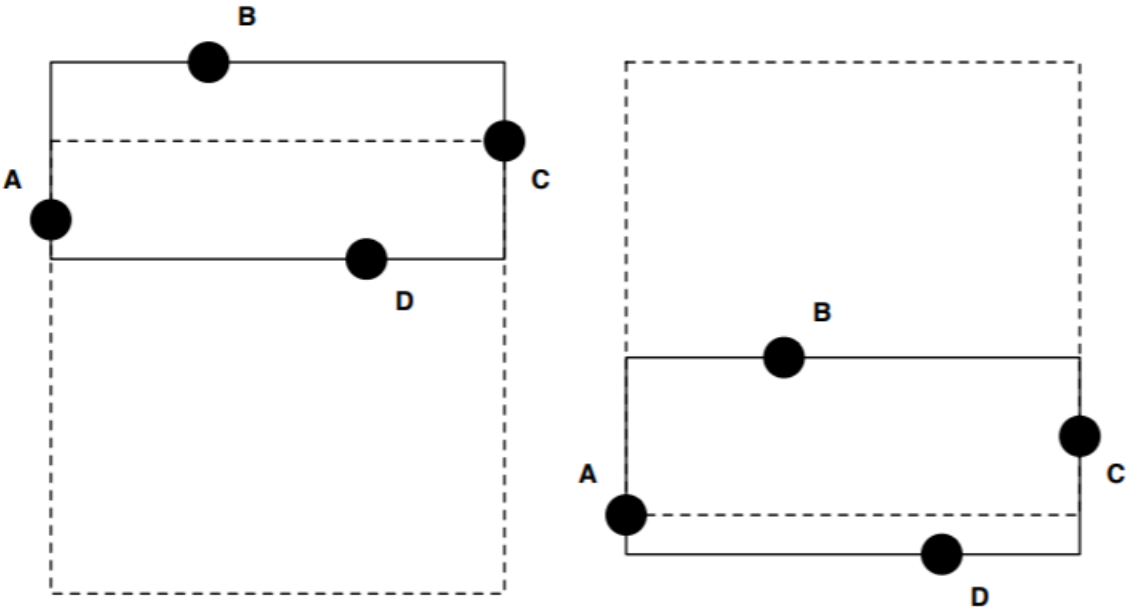
3. Let's consider 3 points that can be shattered.
   Here we are showing how we capture 1, 2, 3 points





Hence there exists an arrangement where 3 points can be shattered.

4. Now let's consider 4 points to be shattered using an axis aligned square.



As we can see there are 4 points arranged in a rectangle. We now need to select 2(A and C) points using the axis aligned square(dotted). In either of the cases that is not possible.

Had this been a rotatable Square this would have been possible.