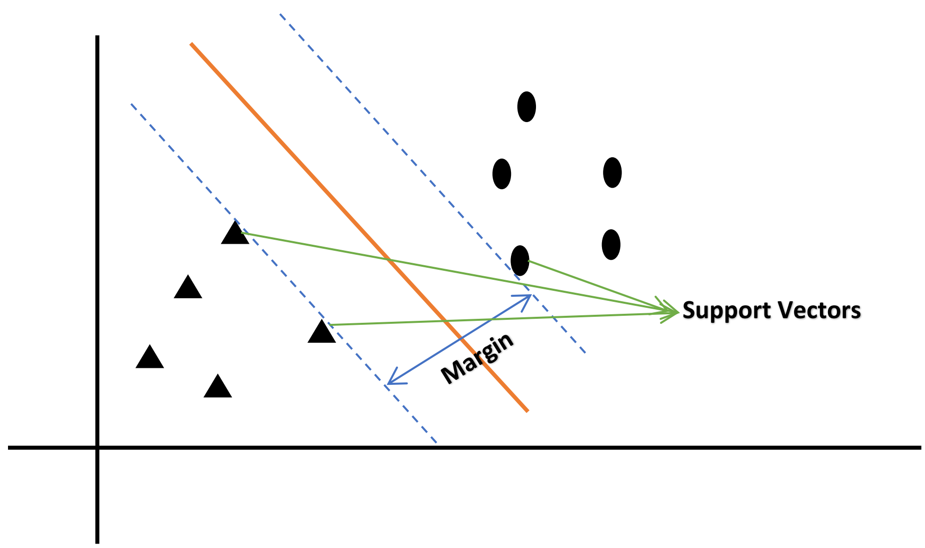
Task 1) What is the margin and support vectors? Q2: How does SVM deal with non-separable data? Q3: What is a kernel? Q4: How does a kernel relate to feature vectors?

→) Margin: - if given a hyperplane, the distance between the hyperplane and closest datapoints. Once we have this distance, if we double it, we call it as margin. We can also define margin as the distance between the observations(datapoints) and the hyperplane.

Support Vector: - Support vectors are the data points which lie closest to the hyperplane. The margin pushes against support the support vector and support vector are the point which touch the boundary of the margin. These points have the direct bearing on the optimum location of the hyperplane.



- 2) In case of linear separable cases, SVM tries to find the hyperplane that maximizes the margin, with the condition that both the classes are classified correctly. But in the case of linear non separable cases the SVM uses two different methods
    - Soft Margin: - This method tries to find a hyperplane to separate but tolerate one or few misclassified data points. In case of soft margins, SVM tolerates a few datapoints to get misclassified and tries to balance the tradeoff between finding a line that maximizes the margin and reduces the misclassification.
    - Kernel Trick & Kernel Function: - Kernel trick utilizes existing features, applies some transformation, and creates new features. It converts data points from low dimensional space to higher dimensional space one by one. The kernel trick uses kernel function to define a high dimensional space. The function is the measure of correlational/distance between two data points in high dimensional space.
- 3) Kernel is a mathematical function that takes the datapoints as input and then transforms it into required form. Kernel function generally transform the data so that nonlinear surface can transform in a linear equation in a higher number of dimension space. It can be the measure of distance between two data points. Different types of kernel functions are:
    - Polynomial kernel function.

- Linear kernel function.
- Gaussian Kernel (RBF) function.
- Sigmoid kernel function.

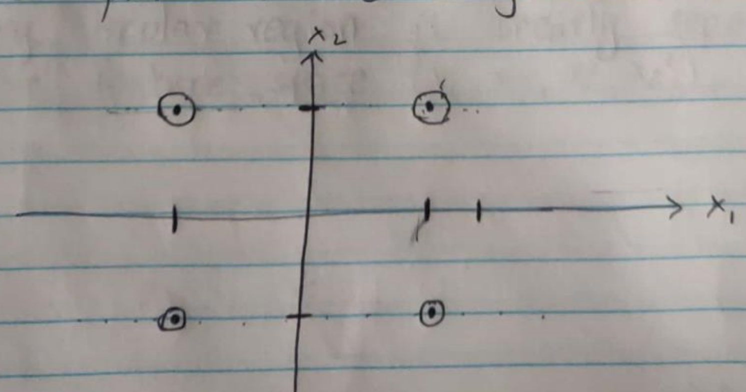
➔ 4) Kernel function has a mathematical function where it acts as the dot product. Kernel function takes input in the original space and returns the dot product of the transformed vectors in the higher dimensional space. Following types of kernel functions are used: -
  - Linear Kernel Function: - It is used when data is linearly separable. It can be determined as follows: -> $k(x_i, x_j) = x_i.x_j$
  - Polynomial Kernel Function: - $k(x_i, x_j) = (1+ x_i.x_j)^d$ ………… d→ the power of polynomial function.
  - Gaussian Kernel Function (RBF): $k(x_i,x_j) = \exp(-(||x_i-x_j||^2)/2\sigma^2)$
  - Sigmoid Kernel Function: - $k(x_i,x_j) = \tanh(\beta_0 x_i^t x_j + \beta_1)$

Task 2
→ [−1, −1] (negative)
[−1, +1] (positive)
[+1, −1] (positive)
[+1, +1] (negative).

The mapping $[x_1, x_2]$ is to $[x_1, x_1x_2]$.
The new feature $[x_1, x_1x_2]$ would be as follows :-

[−1, +1]
[−1, −1]
[+1, −1]
[+1, +1]

∴ The plot of the diagram would be :-



The maximum marginal seperator would be $x_1 x_2 = 0$ & the margin would be 1

Task 2

on one side.

$\therefore$ The total margin would be $= 2$.

Support vector $= [1, 1]$, $[1, -1]$

Task 2

Task 3

$\rightarrow \quad (x_1 - a)^2 + (x_2 - b)^2 - r^2 = 0$

$= \; x_1^2 + a^2 - 2x_1 a + x_2^2 + b^2 - 2x_2 b - r^2 = 0.$

$= \; x_1^2 + x_2^2 - 2(x_1 a + x_2 b) + a^2 + b^2 - r^2 = 0.$

$\therefore$ The weights $\omega = (-2a, -2b, 1, 1).$
by the given feature space $(x_1, x_2, x_1^2, x_2^2).$

We can say that ~~function~~ equation is of the format, $\omega x + b = 0.$

where $\omega = (-2a, -2b, 1, 1)$
$b = a^2 + b^2 - r^2$

$\therefore$ Every circular region is linearly seperable in the feature space $(x_1, x_2, x_1^2, x_2^2).$

Task 3

Task 4

$\rightarrow$ $c(x_1-a)^2 + d(x_2-b)^2 - 1 = 0$

$\Rightarrow c(x_1^2 - 2x_1a + a^2) + d(x_2^2 + b^2 - 2x_2b) - 1 = 0.$

$\Rightarrow cx_1^2 - 2x_1ac + ca^2 + dx^2 + db^2 - 2x_2bd - 1 = 0$

$\Rightarrow cx_1^2 + dx_2^2 - 2x_1ac - 2x_2bd + ca^2 + db^2 - 1 = 0$

$\Rightarrow cx_1^2 + dx_2^2 - 2(x_1ac - x_2bd) + ca^2 + db^2 - 1 = 0$ — (1)

where weight $w = (c, d, 2ac, 2bd, 0)$.

intercept $= ca^2 + db^2 - 1$.

Now the kernel function :-

$k(u,v) = (1 + u \cdot v)^2$

$= 1 + u^2v^2 + 2uv$ — (2)

$\therefore$ consider & substituting the data.

$u = (1, x_1, x_2, x_1^2, x_2^2, x_1x_2)$.

$v = (1, -2ac, -2bd, c, d, 0)$.

$\therefore$ $k(u,v) = (1 + (1, x_1, x_2, x_1^2, x_2^2, x_1x_2) \cdot (1, -2ac, -2bd, c, d, 0)$

$= (1 + (c(x_1-a)^2 + d(x_2-b)^2 - 1))$

$k(u,v) = c(x_1-a)^2 + d(x_2 b)^2$

$= 1$ ... From the main equ

$\therefore$ From $w(x) + b = 0$.

$1 + b = 0.$

$\therefore b = -1$ ... ~~Hence Proved~~

Task 4

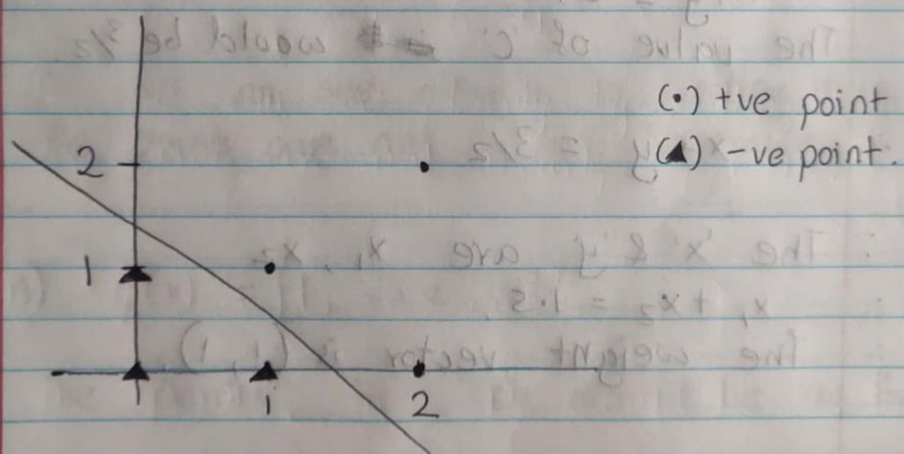
$\therefore$ If we use polynomial kernel, we can prove that the elliptical surface is linearly seperable.

Task 4

Task 5 :-

→ A) Consider the training data :

Class	x_1	x_2
+	1	1
+	2	2
+	2	0
–	0	0
–	1	0
–	0	1



As we can see from the plot, the points are linearly seperable.

Task 5

Task 5 B)

→ The support vectors would be,

$(1,0), (0,1), (1,1)$

$\therefore$ the slope of the line would be '-1'

The equation of line would be :-

$\therefore$ $y = mx + c$

$\because m = -1$

$\therefore$ $y = -x + c$

$x + y = c.$

The value of 'C' would be $3/2$.

$\therefore$ $x + y = 3/2$.

$\because$ The 'x' & 'y' are x_1, x_2

$\therefore$ $x_1 + x_2 = 1.5$.

$\therefore$ The weight vector is $(1, 1)$.

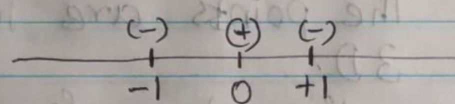
Task 5

Task 6:

→) A) The given dataset:-

Class	x
+	0
−	−1
−	+1

Considering it in 1D.



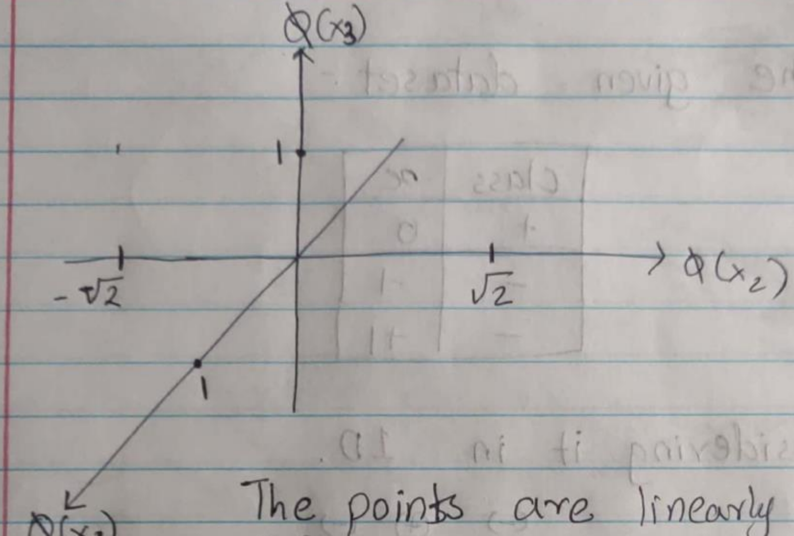
As we can see clearly by plotting it in 1D, the points are not linearly seperable.

B) $\phi(x) = [1, \sqrt{2}x, x^2]$.

∴ The points in 3D would be as follows:-

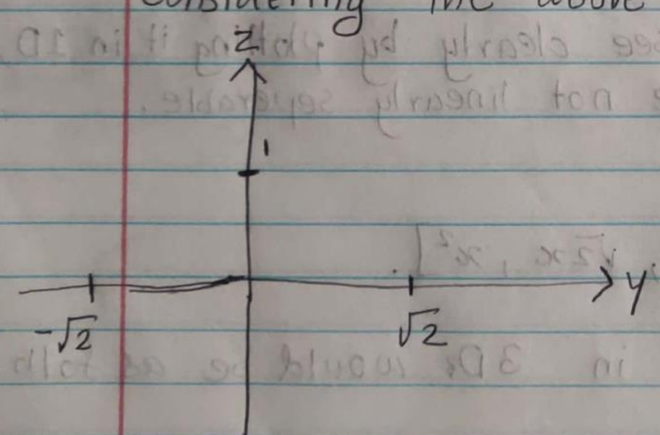
$(1, 0, 0)$
$(1, -\sqrt{2}, 1)$
$(1, \sqrt{2}, 1)$.

Task 6



The points are linearly seperable in 3D.

Considering the above figure in 2D.



The 'x' coordinate is same we will consider only 'y-z' axis.

$\therefore$ The seperating hyperplane would be given by vector $(0, 0, 1)$.

Task 6

Task 7

The best kernel method in my case has been **Linear Kernel Method.**

The link for the code is at the following location.

https://drive.google.com/drive/folders/1ggSEAmx-2kH2iZVBQ5_O4AzVyDsjyvyN?usp=sharing