Support Vector Machines (SVM)

→ supervised Algo that separates data using hyperplanes.

→

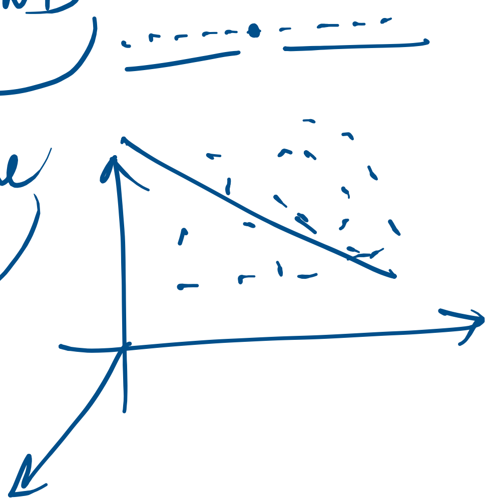
1D → point

2D → line

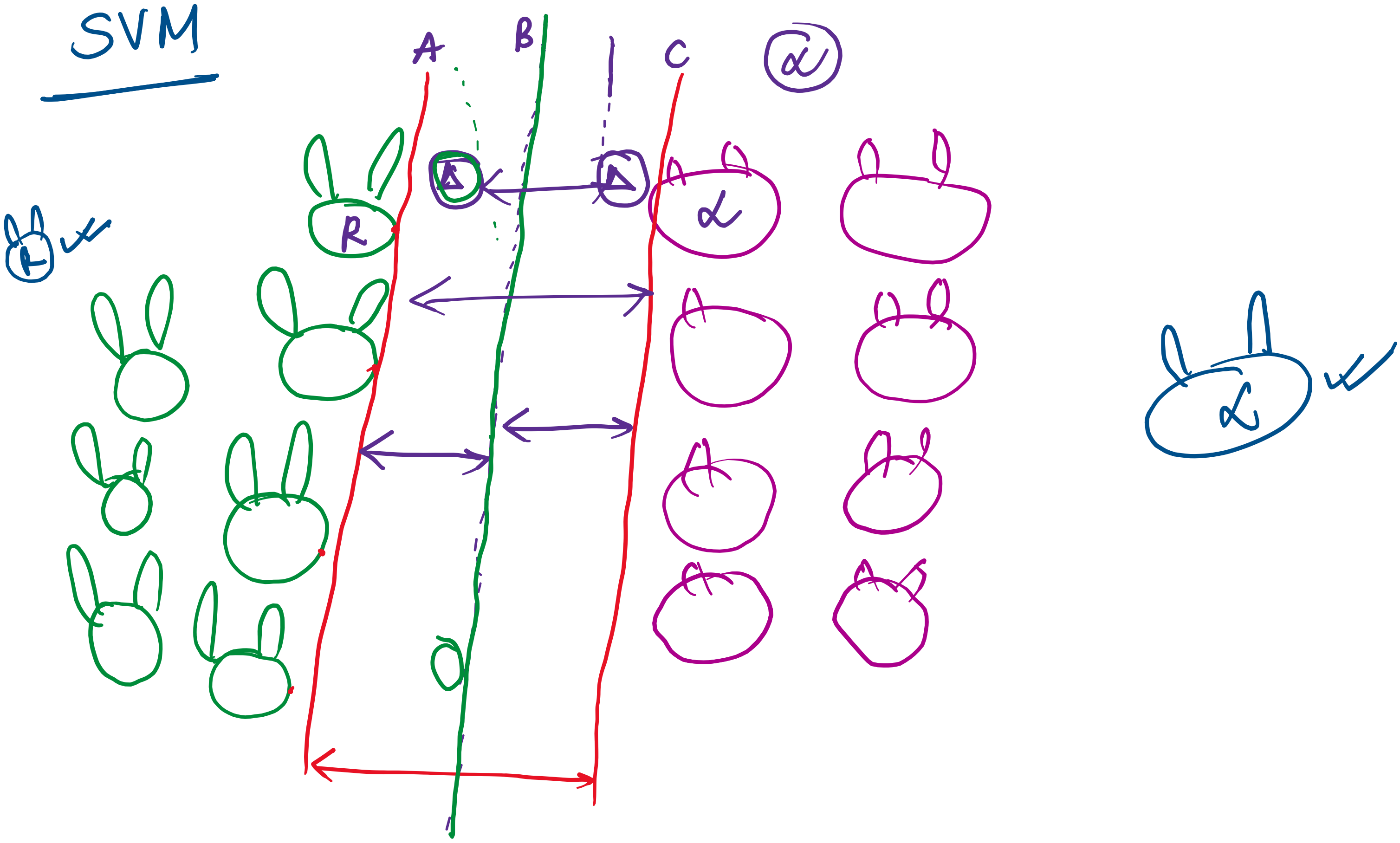
3D → plane

4D 5D ---ND → hyperplane

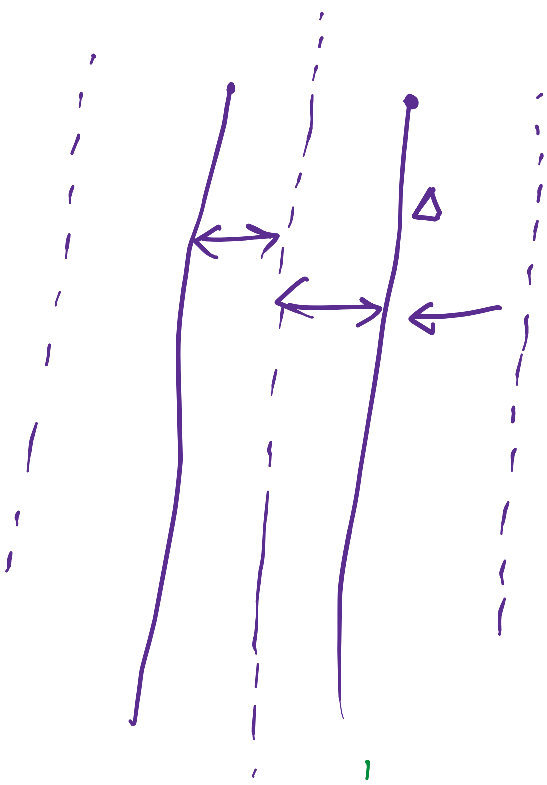
hyperplane.

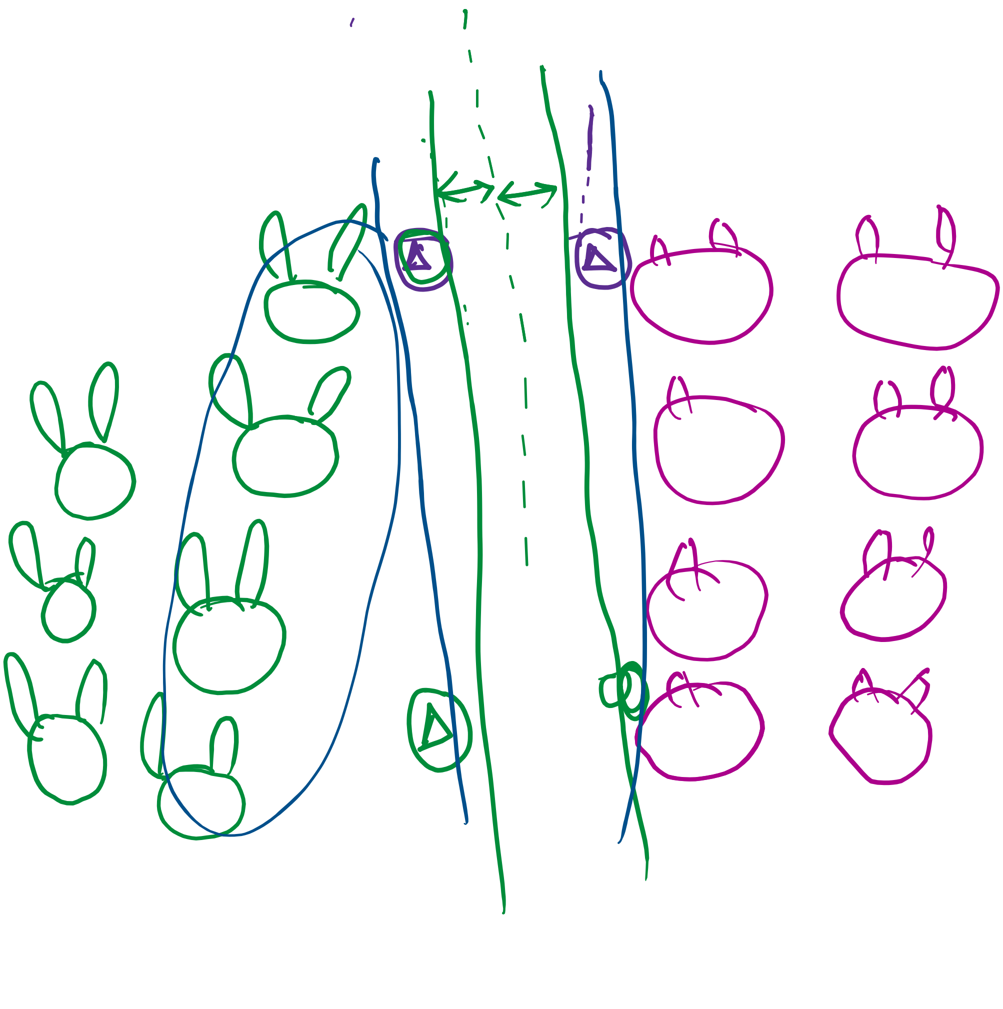


SVM

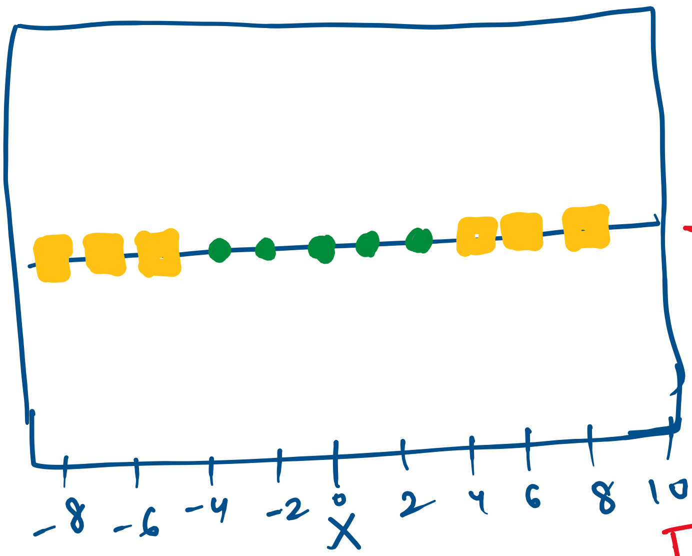


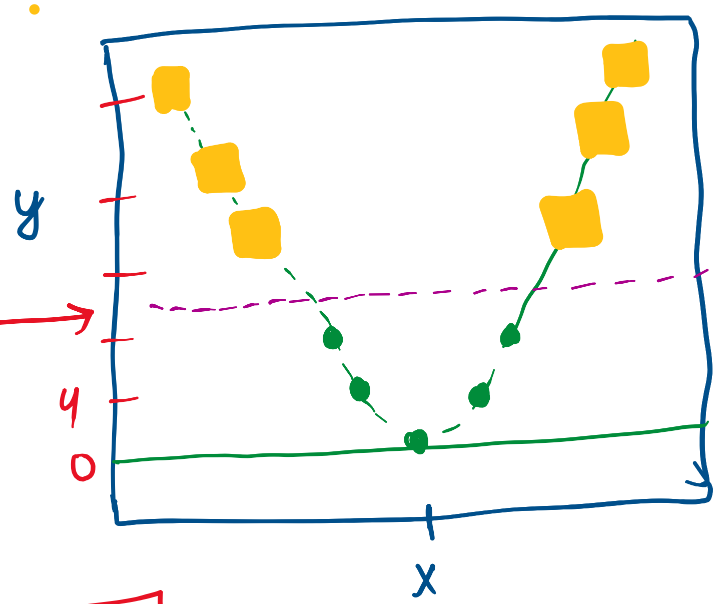
A & C ⇒ support vectors
B ⇒ decision boundary / hyperplane.





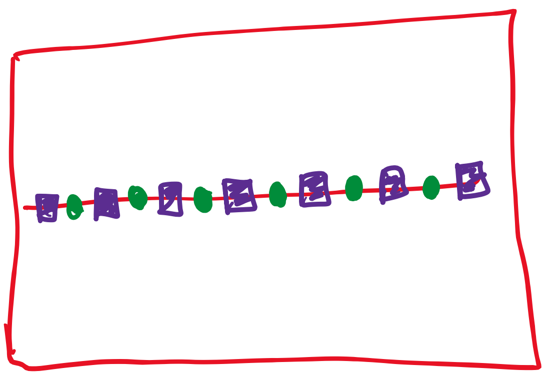
Kernel Trick.

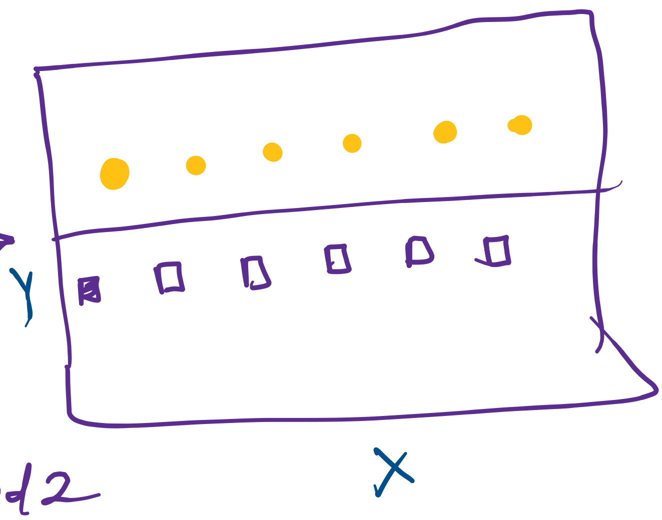




$$\boxed{\phi(x) = x^2}$$

As we can see that the points are not linearly seperable in 1D, but After applying the transformation it became linearly separated.





$$\phi(x) = x \bmod 2$$

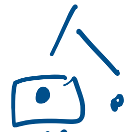
n D $\xrightarrow{\phi(x)}$ $(n+1)$ D

x^n x^2 x

RBF :

Polynomial :

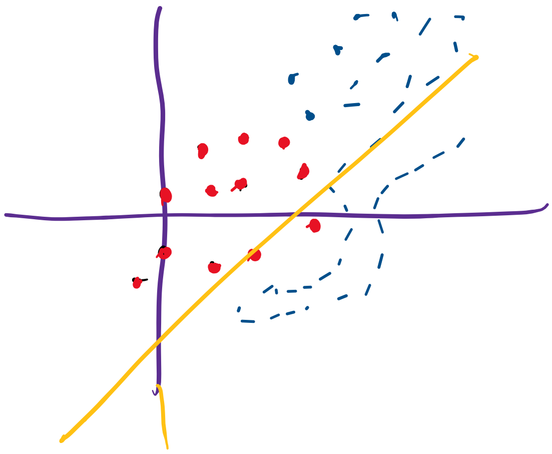
50 — 50 D



100 %

Kernel Trick : Allows us to operate in the original feature space without computing the co-ordinates of data into higher space.

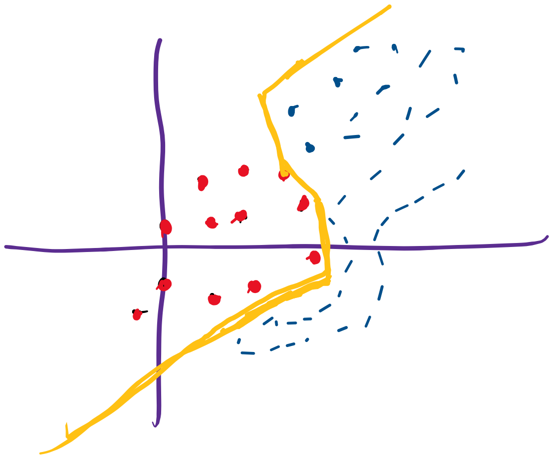
Lets make it a bit more complex.



Some tolerance for outliers.

C ↓

Low Regularization



tolerance = 0
Perfect Partitioning.

C ↑
High Regularization

① Regularization Parameter (C) :-

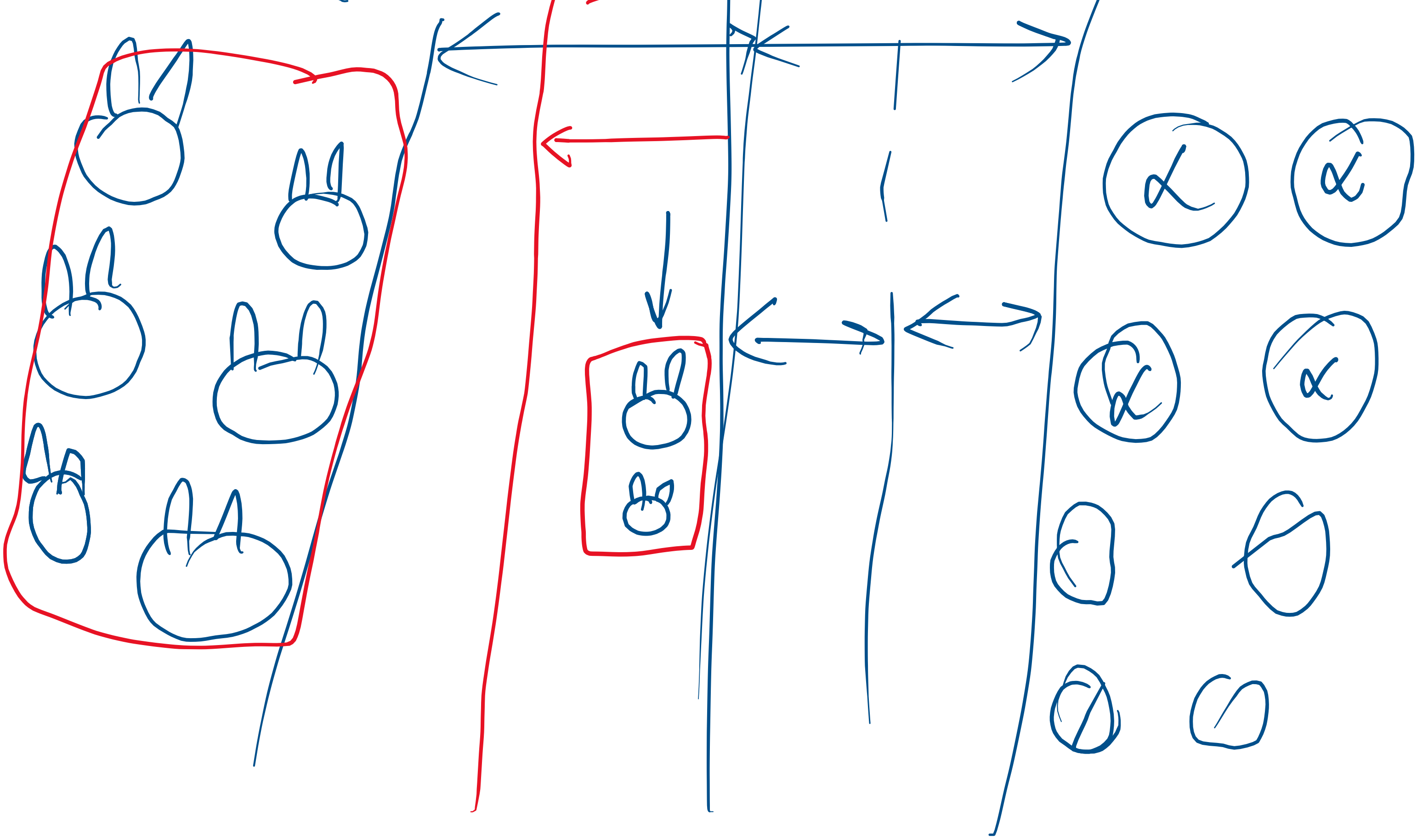
C ↑	model will choose a small margin classifier
C ↓	large margin classifier.

② Gamma:

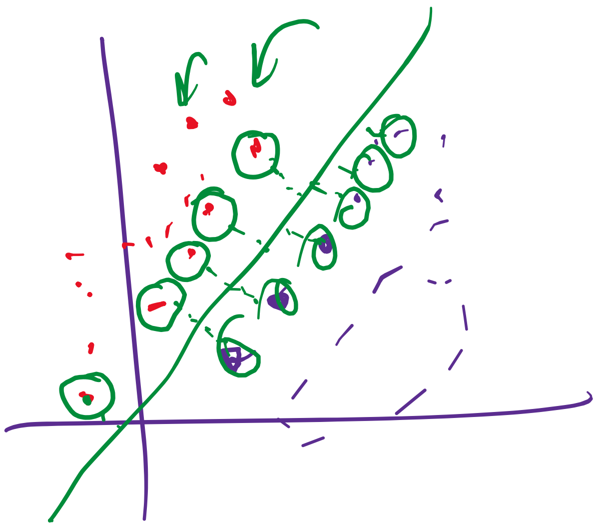
Low gamma value means "far"

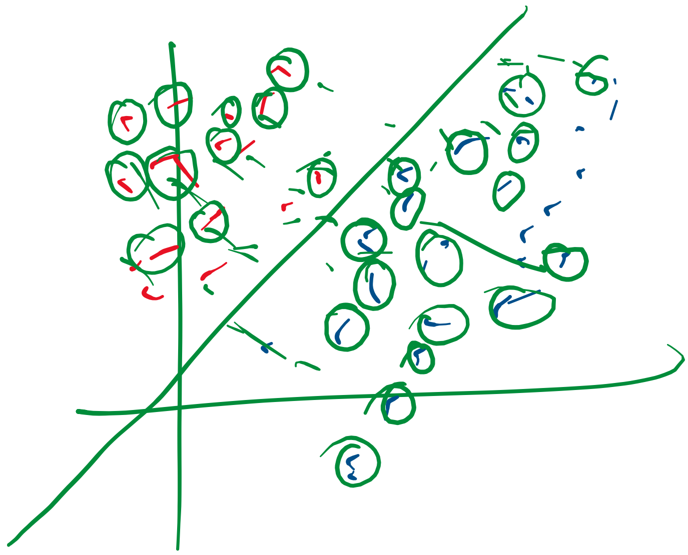
High gamma value meas "only nearby points"

Gama = Low



Gama = High





Low gamma.
↓
far distant Points are also considered.

1. hyperplane
2. Support vector
3. Kernel Trick
4. Regularization Parameter (C) → } Hyperparameter
5. gamma → }
6. Margin

Mathematics (Eqn of hyperplane

1. In 2D (Eqn of hyperplane)

$$\beta_0 + \beta_1 x_1 + \beta_2 x_2 = 0$$

(2) In 3D

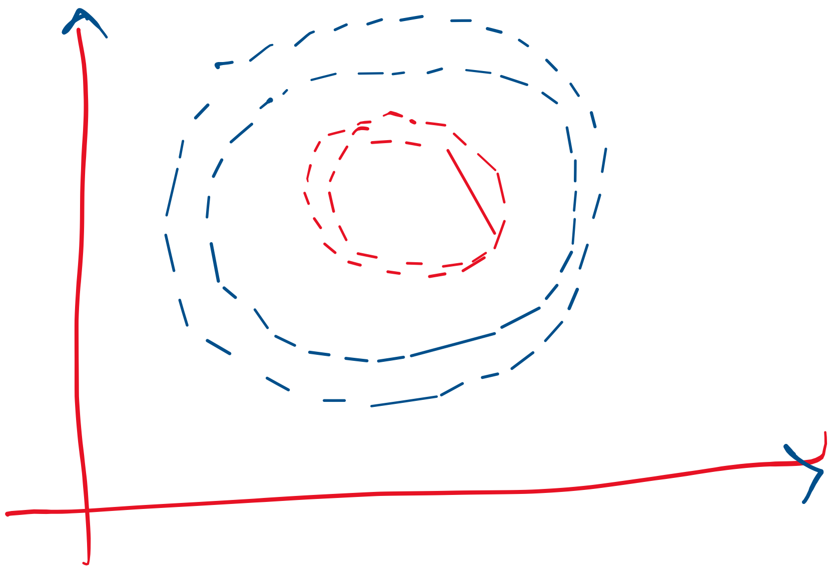
$$\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 = 0$$

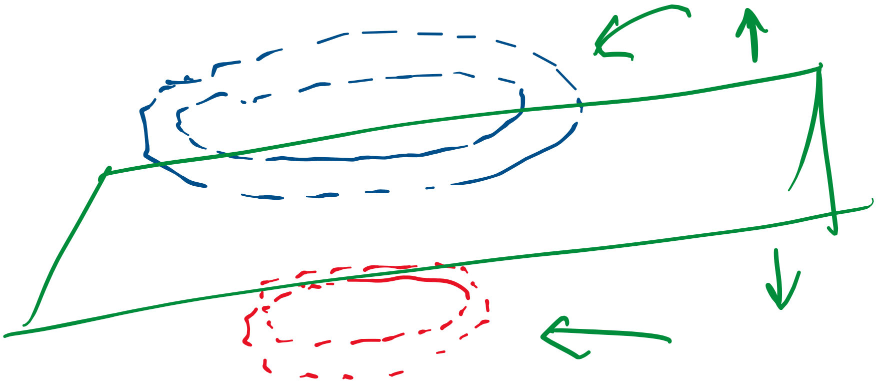
(3) n-D.

$$\beta_0 + \beta_1 x_1 + \cdots\cdots + \beta_n x_n = 0$$

(4) $(\beta_0 + \beta_1 x_1 + \cdots\cdots + \beta_n x_n > 0)$

$(\beta_0 + \beta_1 x_1 + \cdots\cdots + \beta_n x_n < 0)$





EOF>>>