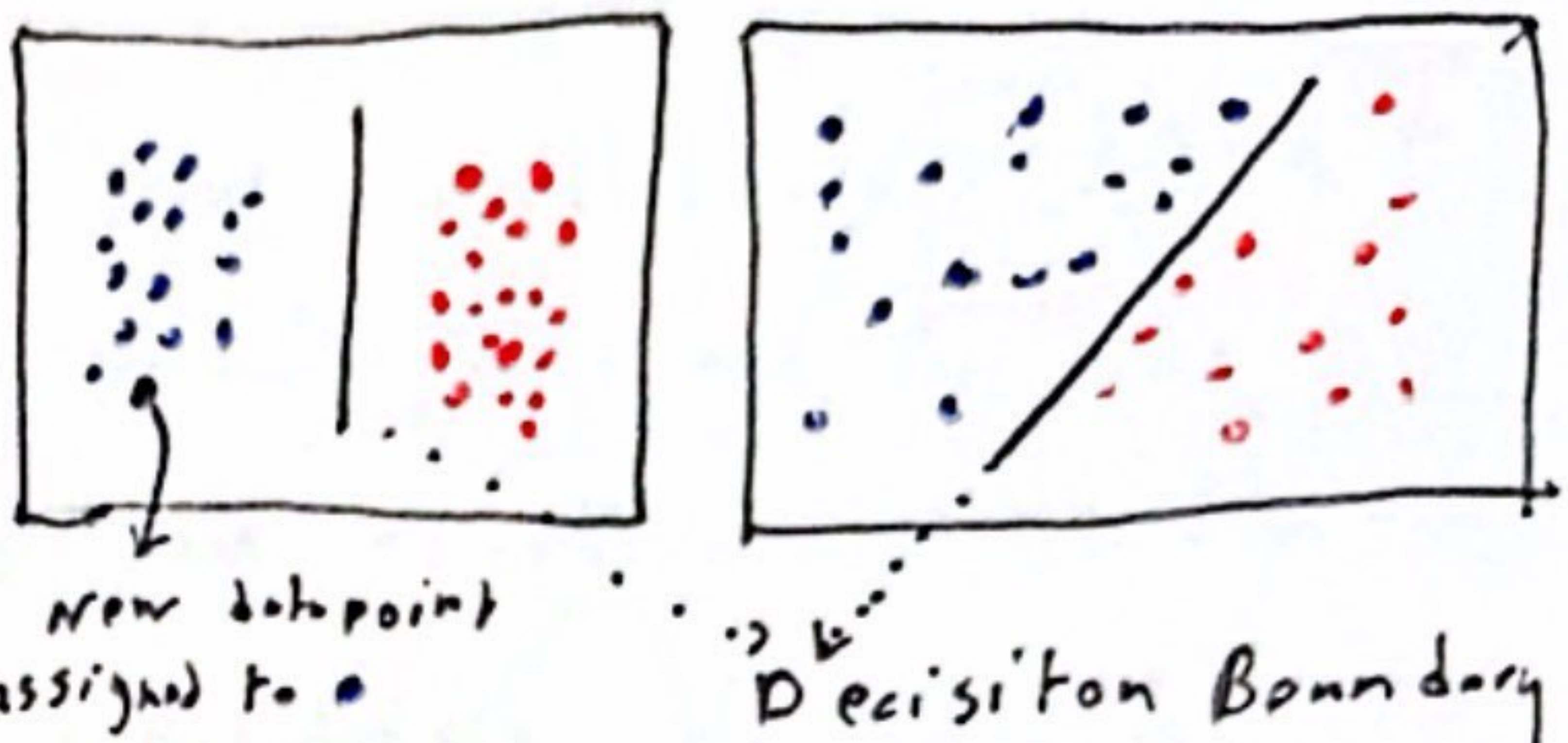
Support Vecter Machine (SVM)

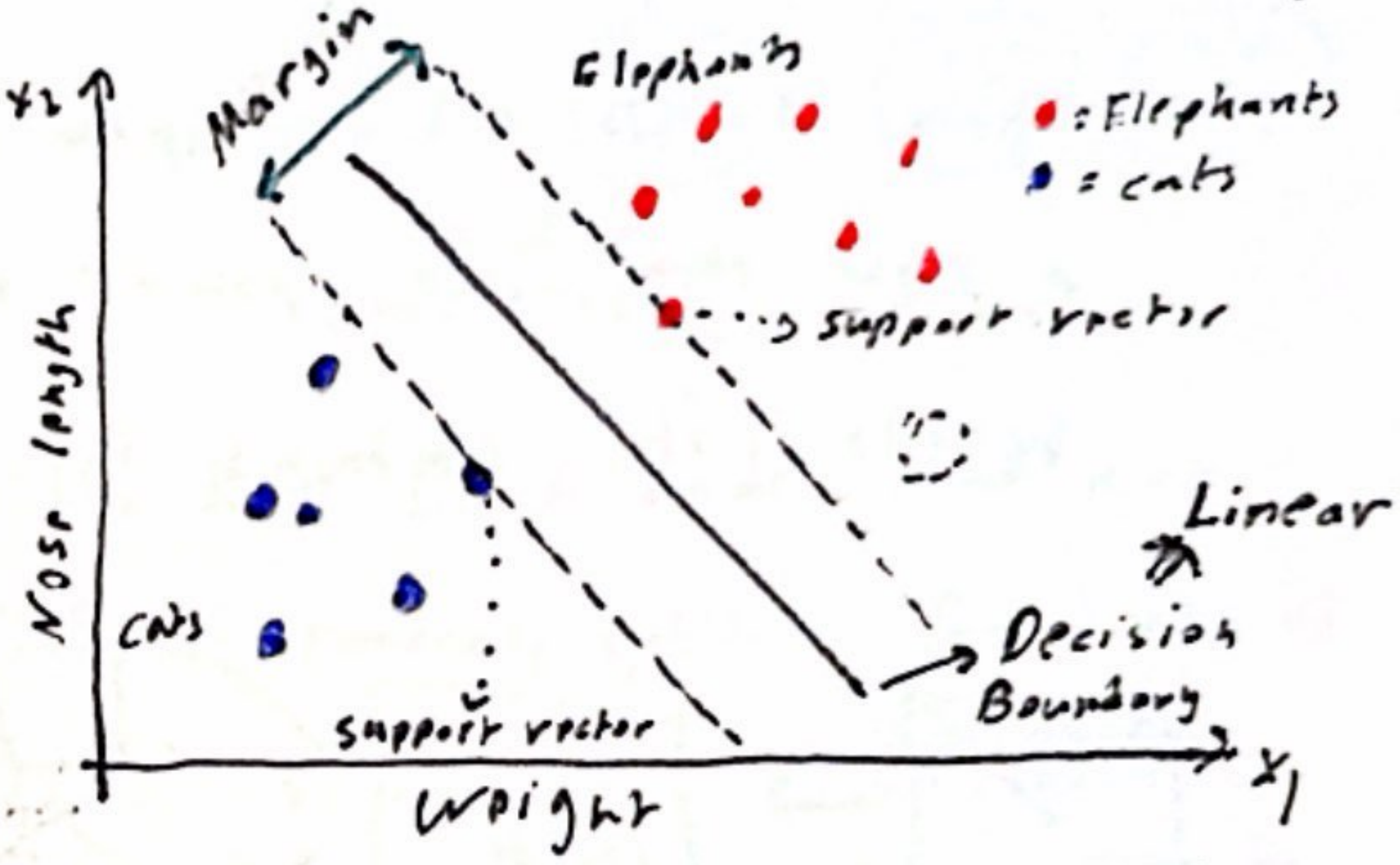
- suprrvised learn
- Classification
- Regretion

1 - The core concept of the algorithm is to draw a decision Boundary Between data points that seperates data points of the training dataset as well as possible

2 ~~A~~ unlike KNN a new data point is assigned to the side of the boundary is falls on the nearest data point does not matter



3 ~~A~~ Ex: lets try to classify animals based weight and len of nose in this case the Decision Bondery is straight the SVM tries to find the line that separates classes with largest margin possible ie maximize space between classes this helps it Genelize well to new data, reduce noise and prevent outliers



4 ~~A~~ The support Vectors are the datapoints that sit on the edge of the margin knowing the SV is enough to classify new Datapoints which makes it memory efficient

5 - if New data point assigned to its class based on the desion boundary
⁚ ⇒ would be classified as Elephant

6 ~~A~~ one Benifit of SVM is that its powerfull in High dimontions ie if # of features is large compared to the data in those cases the decision boundary is complex and called a hyper plane

gamma too ⇒ Depends on Task

-Kernal Function

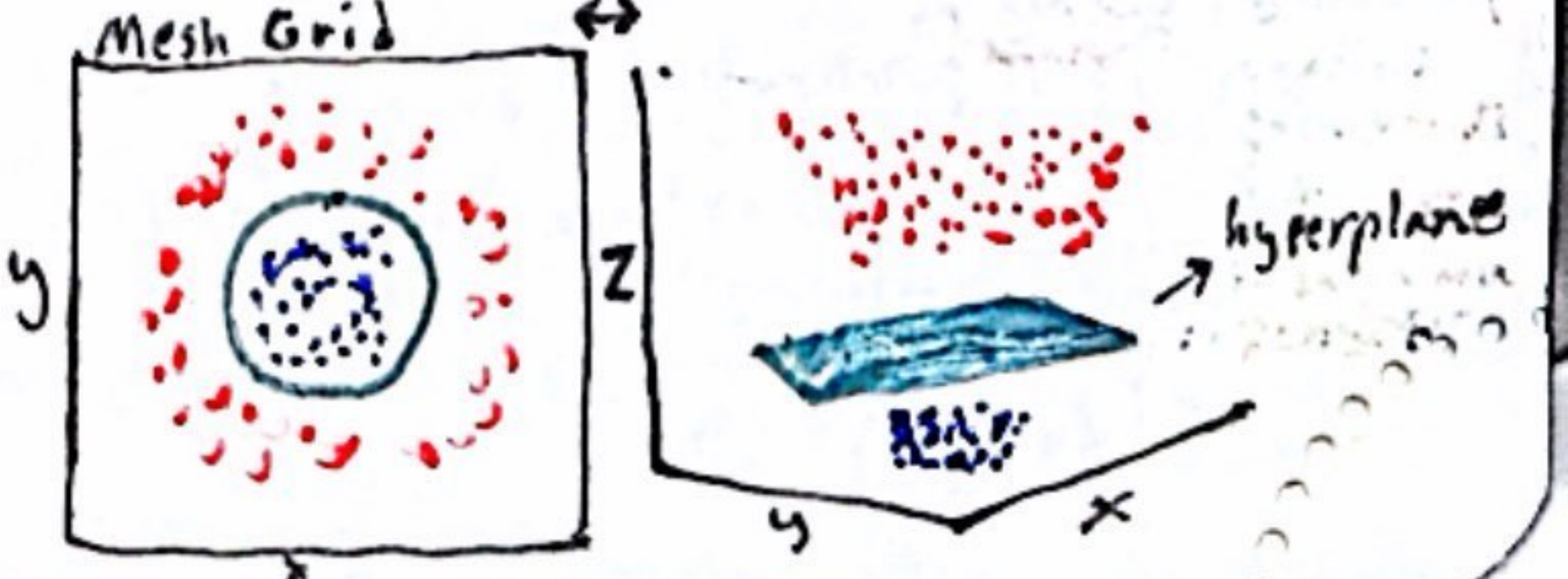
⇒ C = hyperparam, controls how strictly model fits to classify every point, high C = tight fit, less reg = can overfit
Low C = more err tolerance = smother, more gen boundary

7 - another feture of svm's is the use of kernal functions which allow for identification of complex non-linear decition Boundaries
Kernal Functions are a implic way to turn Original Features in to new complex Fetures using kernal trick, basicall make new Fetures for a non-linear Boundary this is called implicit Feature engineering. NN also do this

9 Possible kernal functions in svm
- Dot product
- RBF
- Polynomial
- signoid

/ in non linear: gamma controls influence of each point
high gamma = each point effects small area = tight boundary = overfit & vise versa

8 ~~A~~ Kernal functions Example
— = Non linear Decition Boundary



Mesh Grid
y
Z
hyperplane
y
x