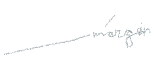
What will happen if the total no of clusters is hugely increased while in unsupervised learning  
Gradient descent formula

Day 3 Class

SVM (Support Vector Machine)  
It is a dual-purpose machine, can be used for regression as well as classification. It is used to convert a regression model into a classification model.

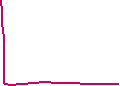
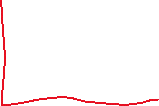
Support Vectors



 The support vectors deal with the nearest datapoints



We map our features in 3D  
We can see the data just levitating



Why use only SVM  
We are increasing the accuracy while doing the classifications

Kernel Function  
Creates an illusion of transforming the datasets into higher dimensions but not actually altering the datasets.  
We are not always using planes for dimensions, (D-1)🡪 planes, we are projecting our data on, so it may create problems  
Major benefit of SVM  
Increasing accuracy by increasing margin distance  
It is letting us use dimensions to enhance classification



🡪These are vectors



We are not mapping, the functions individually, we are taking another function to do it(by mapping it inside the function)



This is the kernel function, it is only mostly used in SVM.

RBF🡪 Gaussian Curve 🡪 It has complex and flexible boundaries  
w.X+b = 0 🡪 Hyperplane  
W= Weight vector  
x= input feature vector  
b = bias term

F(x) = sign(w.x+b) 🡪 Decision Function  
f(x)=+1 🡪 class 1  
f(x)=-1 🡪 class 2

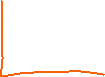
Margin = 2/||w||  
where, ||w|| = square root (w1^2 + w2^2+w3^2+…+wn^2)

We can detect engagement from SVM using regression and it can also be used for spam detection (with the help of classification)

Issues with SVM  
Slower with large dataset  
We need to tune our model (and we changing them is difficult as a very small change can even bring about very large changes).  
  
RBF curve doesn’t have any fixed shape, other curves may have any random shape

KNN  
We check the traits of existing data points  
It is used for classification and regression models (to see the nearest value of the 2 neighbours, the average we get is the nearest neighbour value).  
K-value – we are taking it randomly. Its starts from 1 and goes till the point in which we get the highest accuracy.

We start from one value and we keep on moving. We



K=7 where, k value depends on the total no of pts within a particular given range.

K=3 If we take a small value for k, it takes very less time but it might be very noisy



How it works,  
We will choose then select the number of neighbours (it is basically a hit and trial method, we would be using)  
Calc the distance between the new pt and all the other data points (using Euclidean, Manhattan, etc), so that we get the minimum distance between two points  
We then select the nearest K-neighbour

Manhattan Distance  
The data points are at 90degree, it varies on the type of model and the model you are making/using

Mankowski Distance  
If we use the p value of 1, we get the answer of Manhattan distance and 2 we get the Euclidian distance

KNN vs SVM  
SVM🡪 we are dealing with the margins between the 2 points   
KNN🡪 we are dealing with the geometric shape between the 2 points