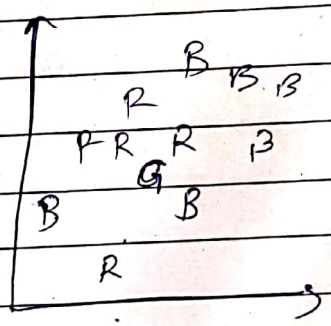
K-NN -> K-nearest neighbours

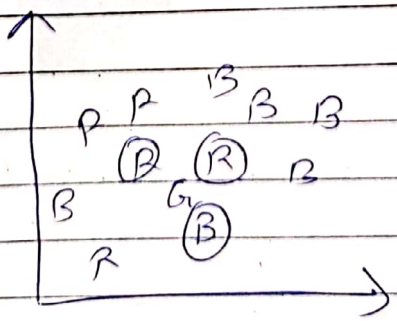
→ KNN is usually a unique algorithm compared to other Supervised Learning algorithms

→ In KNN, the training process is not required, the result will be calculated based on the result of its neighbours.

→ Let's understand this with one example
Suppose we have a data point and we have to predict its color. Let's say new data point is G and let's plot it on graph which already has datapoints of Red and Blue Colour represented by R & B respectively



→ Now from this graph let's consider three nearest neighbours of G which will be



→ From the neighbours above we can see R has its majority so the new data point belongs to Red colour.

→ For KNN training cost is 0 but predicting cost is very high as for every new point the distance should be calculated with every datapoint in the data set

The distance metric is calculated by Eudedian distance.

Let say $u=(u_1, u_2, u_3)$ $v=(v_1, v_2, v_3)$ are two points new distance will be

$$L_2(u,v) = \sqrt{(u_1-v_1)^2 + \ldots + (u_n - v_n)^2}$$

→ whose distance is nearest they are called neighbours.

→ The selection K ~~is~~ is also very crutial. 'K' means how many neighbours should be considered. ~~If~~ say The more the K the more the robust the result will be. But K should be too high also.