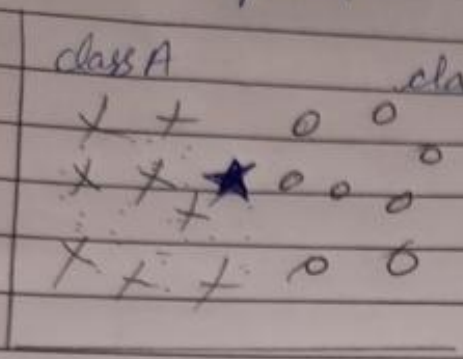
KNN :-

→ It is used for both classification and regression
→ It is a supervised learning algorithm



Let's say you have point ★ which is the query point for which you have to make a prediction with class 1 and class 2.

1] We calculate euclidian distance from ★ to all other data points, the we consider the nearest K points. K denotes how many points are we going to consider near the query point to decide which class it belongs to.
All the work happens at query time, training time is of order 1.

2] All the work happens at query time, training time is of order 1. Query time $O(N)$, $O(Nq)$

3] Non parametric algorithm

4] Most Brute force approach in ML

```
def distance(x1, x2):
    return np.sqrt(sum((x1-x2)**2))
def knn(x, y, queryPoint, k=5):
    val = []
    m = X.shape[0]
    for i in range(m):
        dist = distance(queryPoint, X[i])
        val.append(dist, y[i])
val = sorted(val)
val = val[:k]
max_count = val[1].argmax()
pred = val[0][max_count]
```