ridge_regressor.best_score_

↳ Loss Fxn => Cross-Entropy

# NAIVE BAYES CLASSIFICATION

↳ Based on Bayes Theorem

$$P(A/B) = \frac{P(A) * P(B/A)}{P(B)}$$

---> This model o/ps probability so, it is very small.
So, we need to apply Normaliz-ation

↳ cur $\geq 0.5 \Rightarrow 1$
$< 0.5 \Rightarrow 0$

---> For e.g ↳ $P(Yes/x_i) = 0.13$
$P(No/x_i) = 0.05$

Normalization

$$P(Yes/x_i) = \frac{0.13}{0.13+0.05} = \geq 0.72 \Rightarrow 1$$

★ Use when feature are mostly indepe-ndent.
★ Not great for complex Relationship.
★ Extremly Fast.

---

# KNN (CLASSIFICATION & REGRESSION)

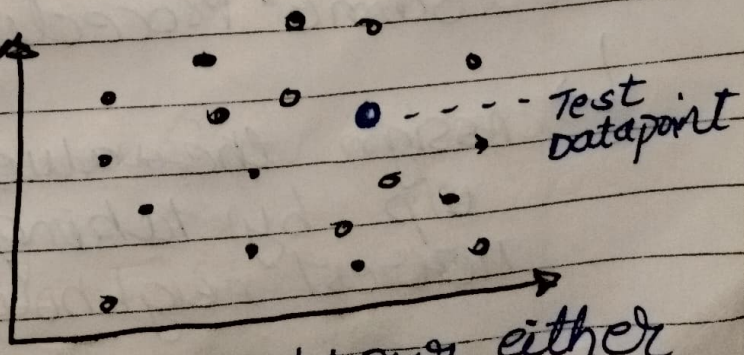
↳ K Nearest Neighbour.

## KNN CLASSIFICATION

Very Easy

i) Select Hyperparameter K.

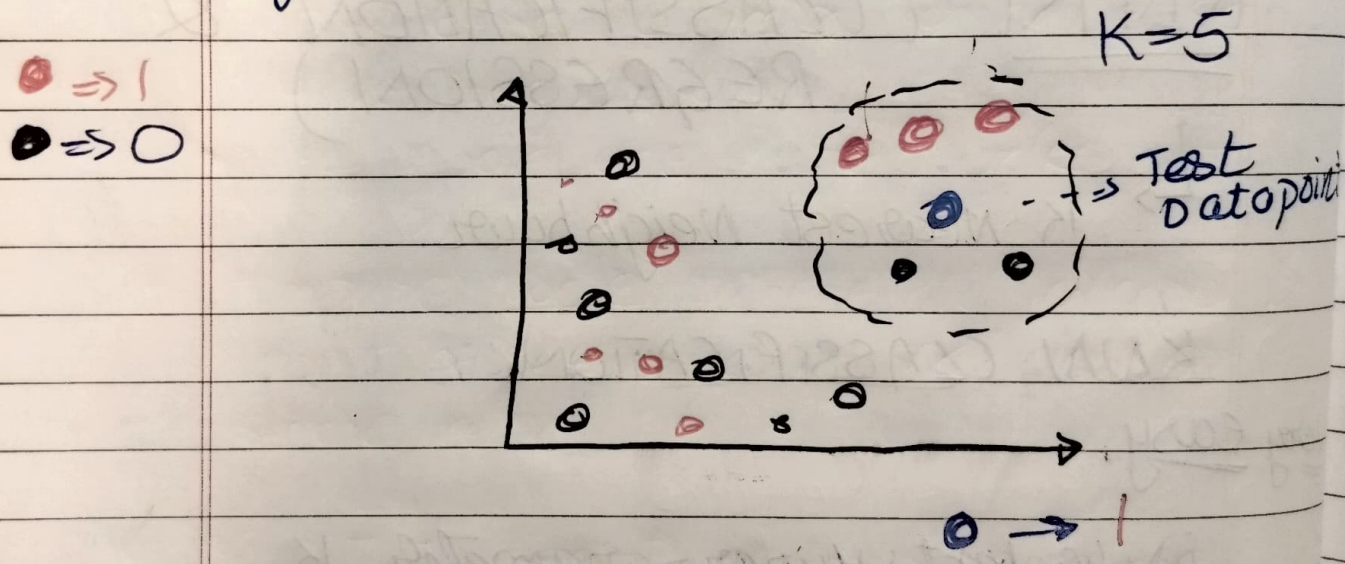
ii) Put the Test Datapoint on the Graph.




iii) Check K Nearest Neighbour either

on the basis of Eucleadian Distance or Manhcatan Distance

(v) From that K nearest neighbour which are of more class, we will assign the label to our datapoint of that class.




## KNN REGRESSION

↳ Same Procedure

↳ Assign the value to new datapoint o/p by taking average of K nearest neighbour o/p.

# Problem with KNN

(I) Should not have outliers.
(II) Should not have imbalanced dataset

## Eucleadian Distance

↳ $\sqrt{(x_2-x_1)^2+(y_2-y_1)^2}$

## Manhattan Distance

↳ $|x_2-y \quad |(x_2-x_1)+(y_2-y_1)|$

## KNN Regression

↳ Loss fxn => ~~MSE~~ No loss fxn

## KNN classification

↳ Don't have formal loss function