Baye's Theorem

$$P(A|B) = \frac{P(B|A) \cdot P(A)}{P(B)}$$

- $P(A|B)$: Posterior
- $P(B|A)$: Likelihood
- $P(A)$: Prior
- $P(B)$: Marginal

$P(A|B)$ → Prob of A given that B has already occured

$$P(A|B) = \frac{P(A \cap B)}{P(B)} \quad -①$$

$$P(B|A) = \frac{P(A \cap B)}{P(A)} \quad -②$$

Conditional Prob ↓ describes the prob of an event occuring given that another event has already occured.

$$P(A|B)\,P(B) = P(A \cap B) \quad -①$$

$$P(B|A)\,P(A) = P(A \cap B) \quad -②$$

RHS is eq so we can compare LHS

$$P(A|B)\,P(B) = P(B|A)P(A) = P(A \cap B)$$

$$\left[P(A|B) = \frac{P(B|A) \cdot P(A)}{P(B)} \right]$$

Hypothesis (Jaha par bhi A dikhta hai usko prob terms me ye bolte hai)

$P(A|B)$

Evidence → (Jaha par bhi B dikhta hai)

$$P(\text{King}|\text{Face}) = \frac{P(\text{Face}|\text{King}) \cdot P(\text{King})}{P(\text{Face})}$$

$$= \frac{1 \cdot 4/52}{12/52} = \frac{1 \cdot 1/13}{3/13} = 1/3$$

Total cards = 52

→ Naive Bayes Classification :-

* Supervised Algo
* Bayes's Theorem
* First understand Bayes theorem

Naive Bayes Classifier Algo

Q) -> Fruit = { yellow, Sweet, Long }

Fruit	Yellow	Sweet	Long	Total
Orange	350	450	0	650
Banana	400	300	350	400
Others	50	100	50	150
Total	800	850	400	1200

$$P(A|B) = \frac{P(B|A) \cdot P(A)}{P(B)}$$

$$P(\text{Yellow}|\text{Orange}) = \frac{P(\text{Orange}|\text{Yellow}) \cdot P(\text{Yellow})}{P(\text{Orange})}$$

$$= \frac{\frac{350}{800} \times \frac{800}{1200}}{\frac{650}{1200}} = 0.53$$

$P(S|O) = 0.69$, $P(L|O) = 0$

$P(\text{Fruit}|\text{Banana}) = 0.75 \times 1 \times 0.87$

$= 0.65$ → max^m prob

$P(\text{Fruit}|\text{Orange}) = 0.53 \times 0.69 \times 0 = 0$

$P(\text{Fruit}|\text{Others}) = 0.33 \times 0.66 \times 0.33$

$= 0.072$

(So, answer is Banana)

SVM

* Supervised Learning
* Used for classification & Reg
* SVM can be used for Face detection, image clasification, text categorization etc.

□ → quadilaterals } Training data
○ → circles } ↓ Labeled data hai

↓

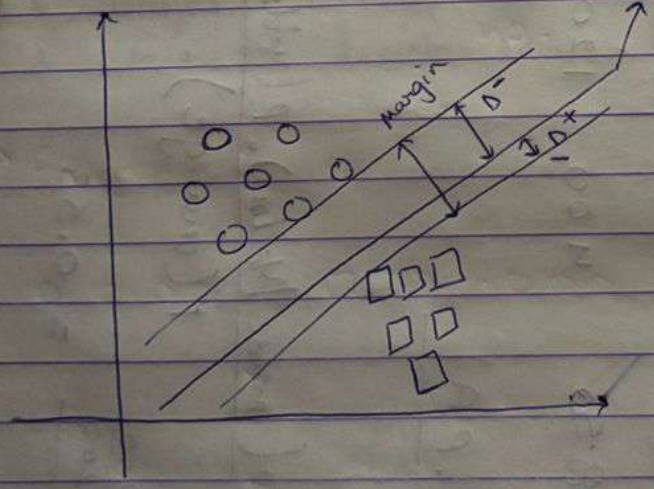
Model Training

↓

Predict ← New data □ ?
○ .

↓

Output



* Inko classes ko seperate rakhna hai toh best way hai inke beech line draw karde.

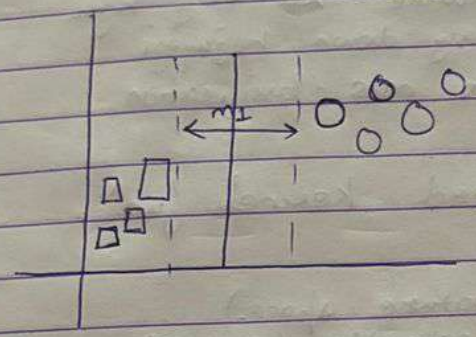
* Margin plays imp role in deciding Hyperplane

Why it is called DB because it decides the whether the new data is □ or ○

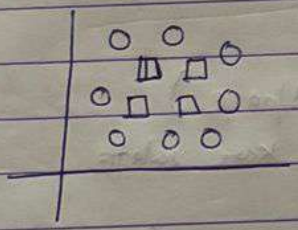
* ½ Jo 2 point consider karke line banai thi unhe bolte hai Support vectors

Q -> Why
the
this

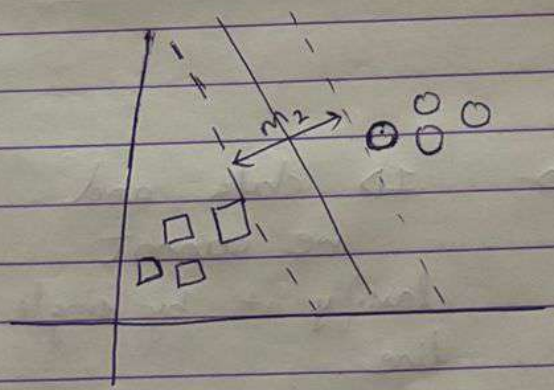
Q-> Why we have make this line to seperate the classes, we can also make lines like this | also !



* When we can just seperate our data by drawing a single line this is called linearly seperable data & on this you can apply linear SVM.



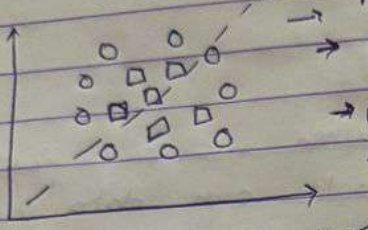
→ Non linear data



* Maximal Margin Hyperplane

→ It means app jo bhi margin select karo uski width sabse jayda honi chahiye, kyuki wo he apki predict achi karega & better accuracy degi.

Non-Linear SVM & Kernel Fun^



→ Non Linear data
→ And we cannot divide this data by a single line.
→ Agar aapne single line se seperate karne ka try kiya then accuracy down hojayegi
→ It leads to mis classification

* So, to clarify it we need kernel fun^.

As Input
1D (Low dimension feature space)
* Issme data non-separable form me hota hai.

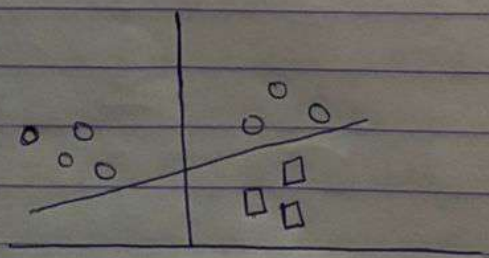
↓ Kernel ↓

As output
HD (High Dimension feature space)
* Issme aap usko seperate kar sakte ho.

→ Example :-

--o-o-oo- --□□□- -ooo- 1D data and we cannot clarify it by single line

↓
So it will goes to Kernel that will change it into HD - that means into 2D data.



* Some
which
changes

* Sometimes we have to give 2D data which is non-seperable then Kernel fun^ changes into 3-D plane.