

### 3. Naive Baye's Alforithms

- Let's say we have a dice ,so what is the probability of getting 1 ?

**P(1) = 1/6** , likewise the probability of getting 4 is the same **P(4) = 1/6** as these two are independent events.

- Now, let's discuss what is dependent event,

For example, we have 3 red balls and 2 green balls in a box, what is the probability of getting red ball.

$$P(\text{red ball}) = 3/5$$

Now, after this event what is the probability of getting green ball,

$$P(\text{green ball}) = 2/4 = 1/2$$

Because after the first event , there are only 4 balls left so the probability of green ball is **1/2** .

Now, **P(Red and Green) = P(red) \* P(Green / red)** , This is called **Conditional Probability**.

- Also, We can say that **P(Red and green) = P(green and red)**

$$P(\text{red}) * P(\text{Green} / \text{red}) = P(\text{Green}) * P(\text{Red} / \text{green}) ,$$

So , **P(Green / red) = P(Green) \* P(Red / green) / P(red)** , This is called **Baye's Theorem**

Let's say in our dataset , we have features  $x_1, x_2, x_3, x_4, \dots, x_n$  and output  $Y$

As per the baye's theorem , we can say that

$$P(Y/x_1, x_2, x_3, x_4, \dots, x_n) = P(y) * P(x_1, x_2, x_3, x_4, \dots, x_n / Y) / P(x_1, x_2, x_3, x_4, \dots, x_n)$$

Now , if i extend it , how it looks like

$$P(Y/x_1, x_2, x_3, x_4, \dots, x_n) = \frac{p(Y) * P(x_1/y_1) * P(x_2/y_2) * P(x_3/y_3) * P(x_4/y_4) * \dots * P(x_n/y_n)}{P(x_1) * P(x_2) * P(x_3) * P(x_4) * \dots * P(x_n)}$$

Let's take an example of binary classification, we have output **yes** or **no**

$$P(\text{Yes} / X_i) = \frac{p(Y) * P(x_1/\text{yes}) * P(x_2/\text{yes}) * P(x_3/\text{yes}) * P(x_4/\text{yes}) * \dots * P(x_n/\text{yes})}{P(x_1) * P(x_2) * P(x_3) * P(x_4) * \dots * P(x_n)}$$

Here, We will ignore the bottom as its constant ,SO

$$P(\text{Yes} / X_i) = P(Y) * P(x_1/\text{yes}) * P(x_2/\text{yes}) * P(x_3/\text{yes}) * P(x_4/\text{yes}) * \dots * P(x_n/\text{yes})$$

**Likewise , for No**

$$P(\text{No} / X_i) = \frac{p(n) * P(x_1/\text{no}) * P(x_2/\text{no}) * P(x_3/\text{no}) * P(x_4/\text{no}) * \dots * P(x_n/\text{no})}{P(x_1) * P(x_2) * P(x_3) * P(x_4) * \dots * P(x_n)}$$

Here, We will ignore the bottom as its constant

$$P(\text{No} / X_i) = P(N) * P(x_1/\text{no}) * P(x_2/\text{no}) * P(x_3/\text{no}) * P(x_4/\text{no}) * \dots * P(x_n/\text{no})$$

If we do  $p(\text{yes}/X_i) = P(\text{No}/X_i)$  , that's why we removed constant as it's same in both.

Let's consider that we got  $P(\text{Yes} / X_i) = 0.13$  and  $P(\text{No} / X_i) = 0.05$  , now we know in binary classification that if output is more than or equal to **0.5** we consider it as **1** , and if output is less than **0.5** We consider it as **0**.

Which one Should we consider as a output in above example 0.13 or 0.05 as we will get both 0, here we will use normalization,

$$P(\text{Yes} / X_i) = 0.13 / (0.13 + 0.05) = 0.72 = 72\%$$

$$P(\text{No} / X_i) = 1 - 0.72 = 28\%$$

Let's say we have tennies dataset

Live Day 3- Discussing KNN And Naive Baye's Machine Learning Algorithms

Scibe Ink

Untitled (3/3)

DATA SET

Binary Class

$x_1$   
Outlook

Day	Outlook	Temperature	Humidity	Wind	Play Tennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

Y<sub>0</sub>    N<sub>0</sub>    P(Y<sub>0</sub>)    P(N<sub>0</sub>)

Sunny    2    3    2/9    3/5

Overcast    4    0    4/9    0/5

Rain    3    2    3/9    2/5

Total    9    5

1:40:32 / 2:02:35

Scroll for details

Live Day 3- Discussing KNN And Naive Baye's Machine Learning Algorithms

Scibe Ink

Untitled (3/3)

D11    Sunny    Mild    Normal    Strong    Yes

D12    Overcast    Mild    High    Strong    Yes

D13    Overcast    Hot    Normal    Weak    Yes

D14    Rain    Mild    High    Strong    No

Temperature

Rain    3    2

Total    9    5

PLAY

Yes    No    P(Y)    P(N)

Hot    2    2    2/9    2/5

Mild    4    2    4/9    2/5

Cold    3    1    3/9    1/5

Total    9    5

Yes    No    P(Y<sub>0</sub>)    P(N<sub>0</sub>)

Total    9    5    9/14    5/14

1:43:45 / 2:02:35

Scroll for details

Here, we will consider our features outlook, temperature, Humidity, wind as a  $x_1, x_2, x_3, x_4$  respectively.

We make table for each feature as shown in above images then we do testing on test data.

Let's take test data the you have Outlook feature **Sunny** and temperature feature **Hot** then what would be my Output either **yes** or **No**

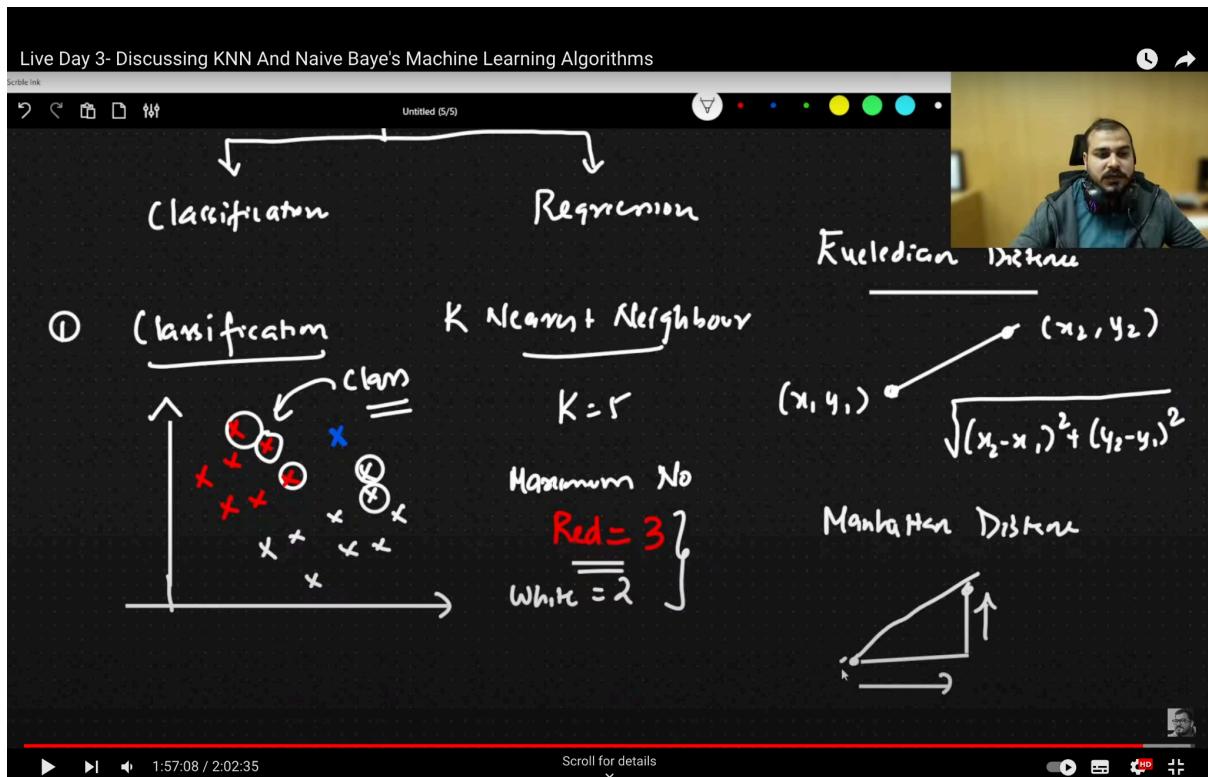
Read this Note

$$P(\text{yes} / \text{sunny}, \text{Hot}) = P(\text{Yes}) * p(\text{Sunny}/\text{yes}) * P(\text{Hot}/\text{yes})$$

For No ,

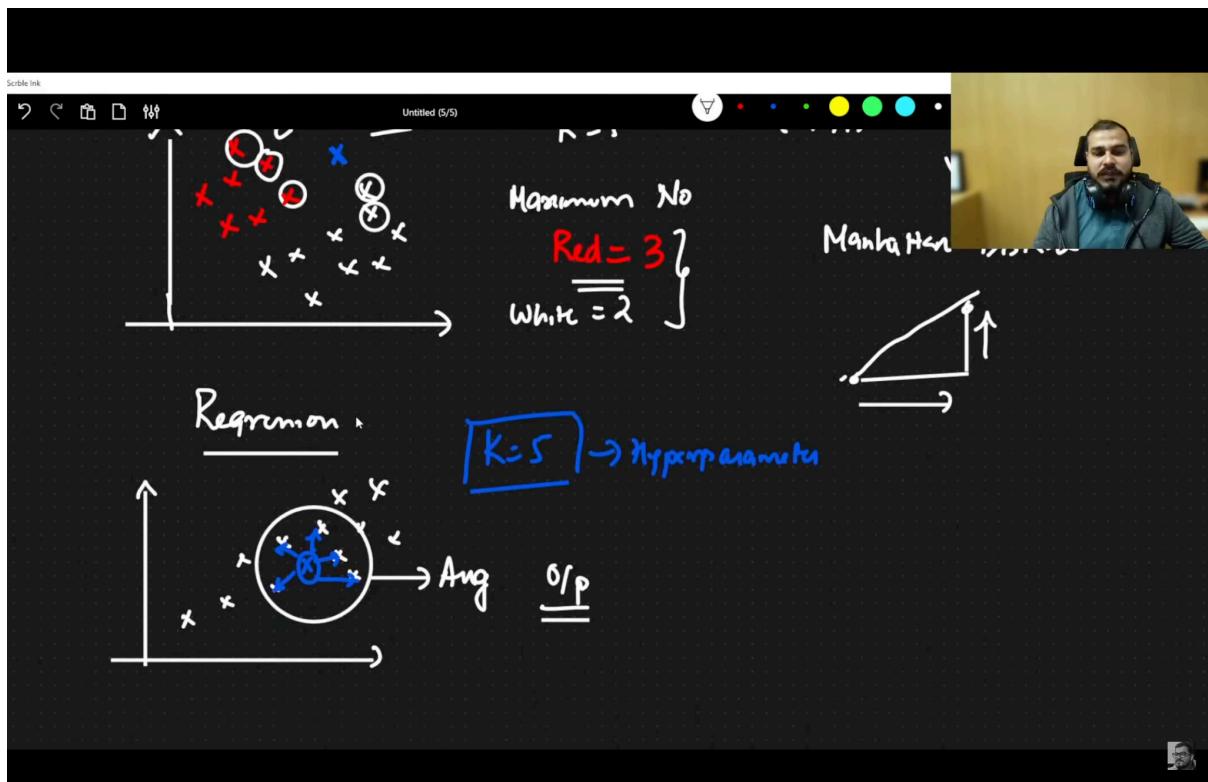
$$P(\text{no} / \text{sunny}, \text{Hot}) = P(\text{no}) * p(\text{Sunny}/\text{no}) * P(\text{Hot}/\text{no})$$

## 4. KNN Algorithm



KNN for Classification

- In classification, our new data will be decided from the nearest neighbour's as shown in above image.
- So, how we calculate the distance from new data to actual points. Here, we will use two methods for that first and foremost is **Eucledian Distance** and Secondly, **Manhattan Distance**.



KNN for Regression