3. Naive Baye’s Alforithms

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

P(1) = ⅙ , likewise the probability of getting 4 is the same **P(4) =** ⅙ as these two are independent events.

- Now, let’s discuss what is dependent event,
  For example, we have 3 reb balls and 2 green balls in a box, what is the probability of getting red ball.

  P(reb ball) = ⅗

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

  P(green ball) = 2/4 = ½

  Because after the first event , there are only 4 balls left so the probability of geen 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(red) * P(Green / red) = P(Green) * P(Red / 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 x1,x2,x3,x4,...,xn and output Y

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

P(Y/ x1,x2,x3,x4,...,xn) = P(y) * P(x1,x2,x3,x4,...,xn / Y) / P(x1,x2,x3,x4,...,xn)

Now , if i extend it , how it looks like

P(Y/x1,x2,x3,x4,...,xn) = $\frac{p(Y) * P(x1/y1)* P(x2/y2)* P(x3/y3)* P(x4/y4)* ...* P(xn/yn)}{P(x1) * P(x2)*P(x3)*P(x4)* ... *P(xn)}$

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

P(Yes/ Xi) = $\frac{p(Y) * P(x1/yes)* P(x2/yes)* P(x3/yes)* P(x4/yes)* ...* P(xn/yes)}{P(x1) * P(x2)*P(x3)*P(x4)* ... *P(xn)}$

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

P(Yes / Xi) = P(Y) * P(x1/yes) *P(x2/yes) *P(x3/yes) *P(x4/yes) * …P(xn/yes)

Likewise , for No

P(No / Xi) = $\frac{p(n) * P(x1/no)* P(x2/no)* P(x3/no)* P(x4/no)* ...* P(xn/no)}{P(x1) * P(x2)*P(x3)*P(x4)* ... *P(xn)}$

Here, We will ignore the bottom as its constant

P(No/ Xi) = P(N) * P(x1/no) *P(x2/no) *P(x3/no) *P(x4/no) * …P(xn/no)

If we do p(yes/Xi) = P(No/xi) , that’s why we removed constant as it’s same in both.

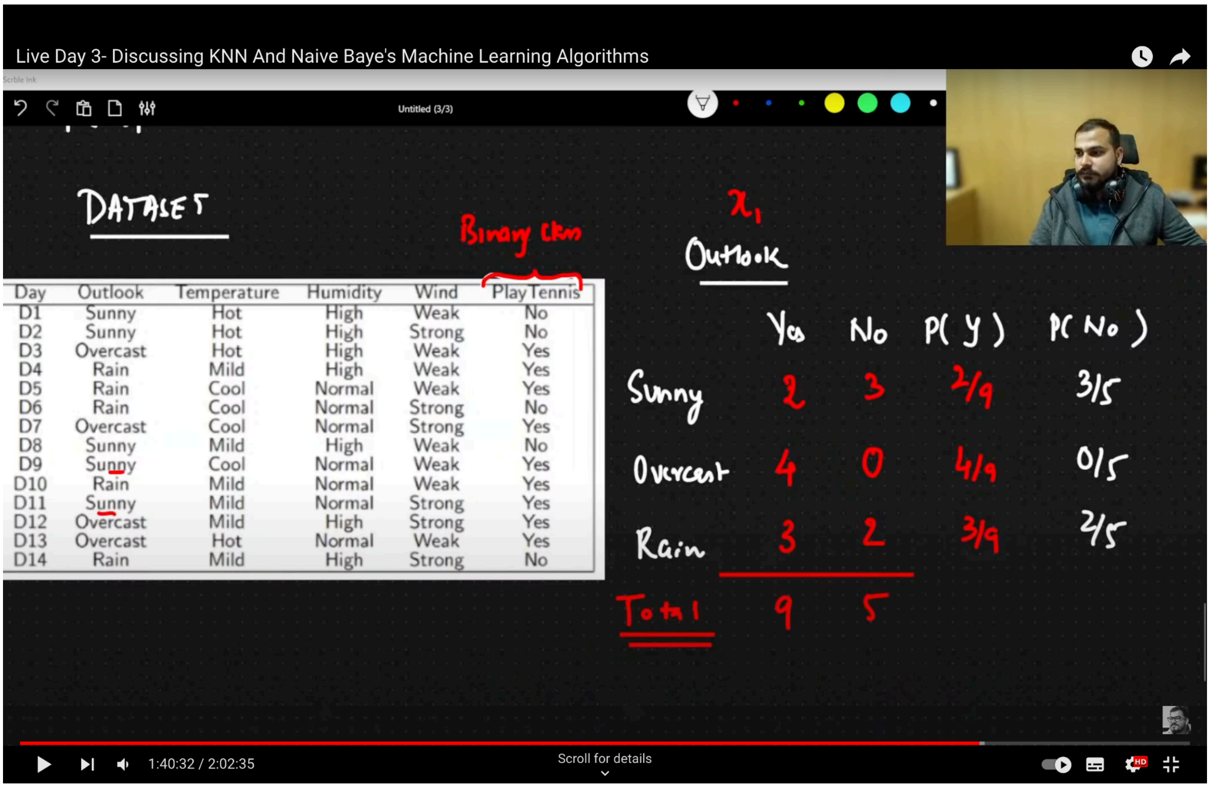
Let’s consider that we got **P(Yes / Xi) = 0.13 and P(No / Xi) = 0.05 ,** now we know in binary classification that if output is more than or eual to **0.5** we consider it as **1 , and** if output is less than **0.5** We consider it as **0.**

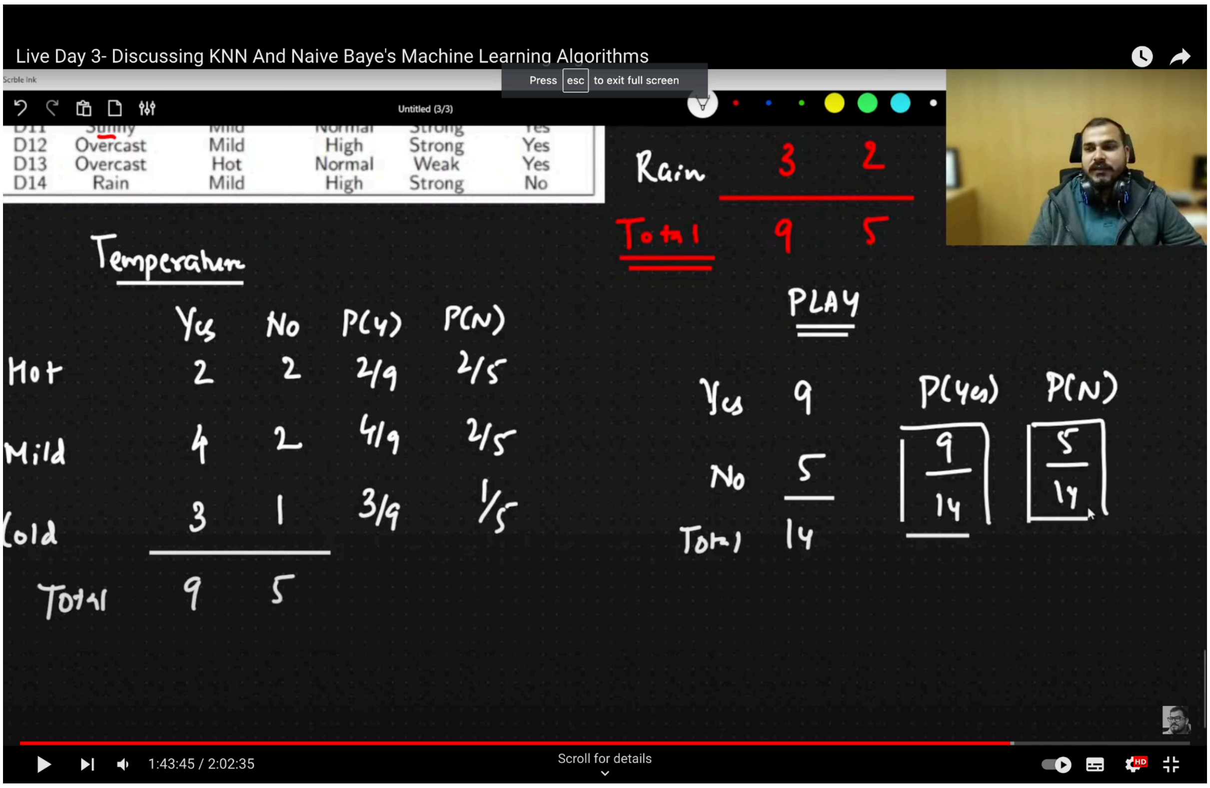
Which one Should we consider as a output in above example 0.13 or 0.05 as we will get both 0, here we will use normalization,

P(Yes / Xi) = 0.13 / 0.13 + 0.05 = 0.72 = 72%

P(No / Xi) = 1- 0.72 = 28%

Let’s say we have tennies dataset





Here, we will consider our features outlook, temperature, Humidity, wind as a x1,x2,x3,x4 respectively.

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

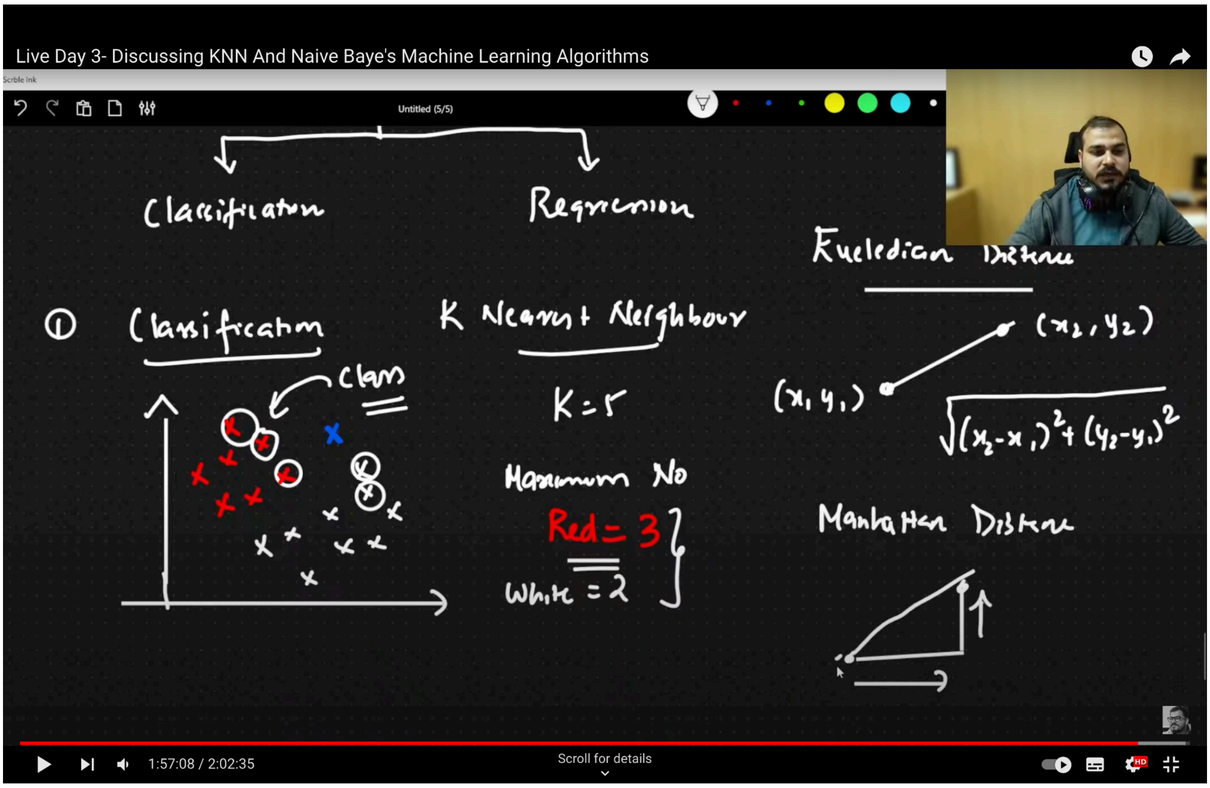
Let’s take test data the you have Outlook feature **Sunny and** temperature feature **Hot** then what would be my Output either **yes** or **No**

Read this **Note**
P(yes / sunny,Hot) = P(Yes) * p(Sunny/yes) * P(Hot/yes)

For No ,

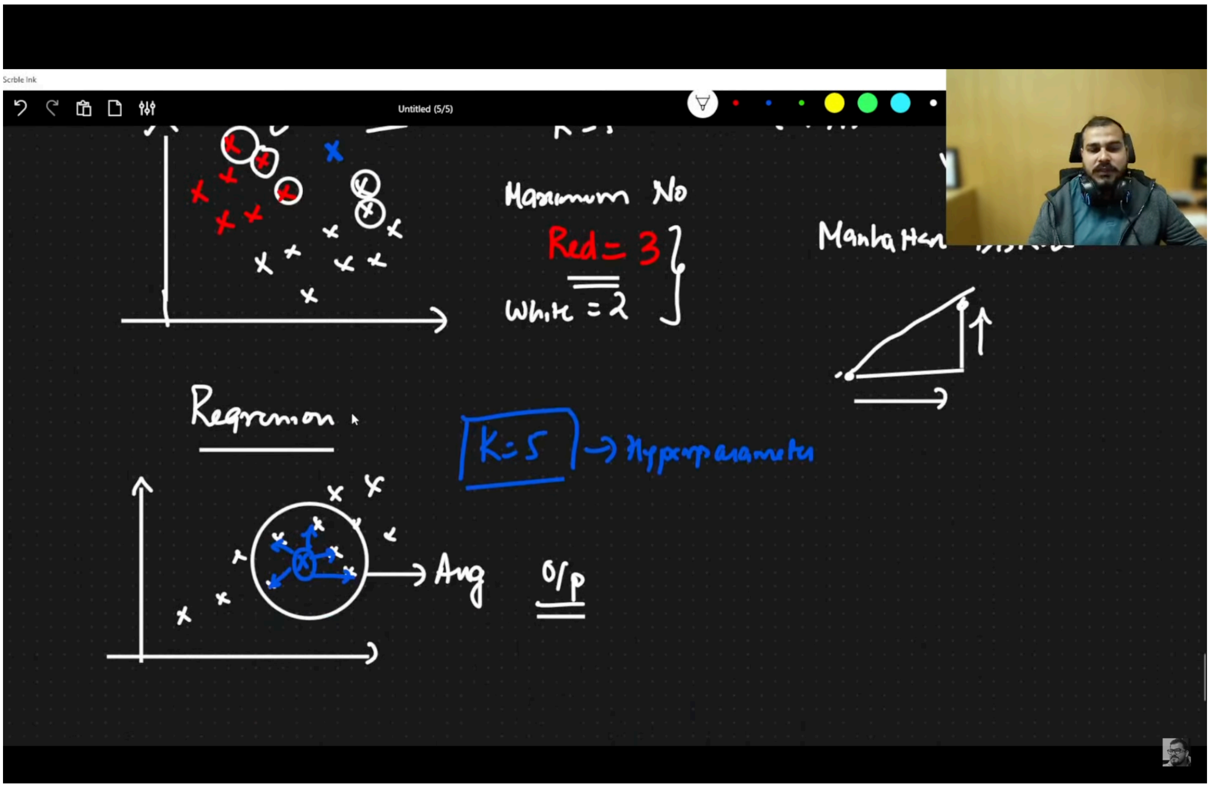
P(no / sunny,Hot) = P(no) * p(Sunny/no) * P(Hot/no)

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