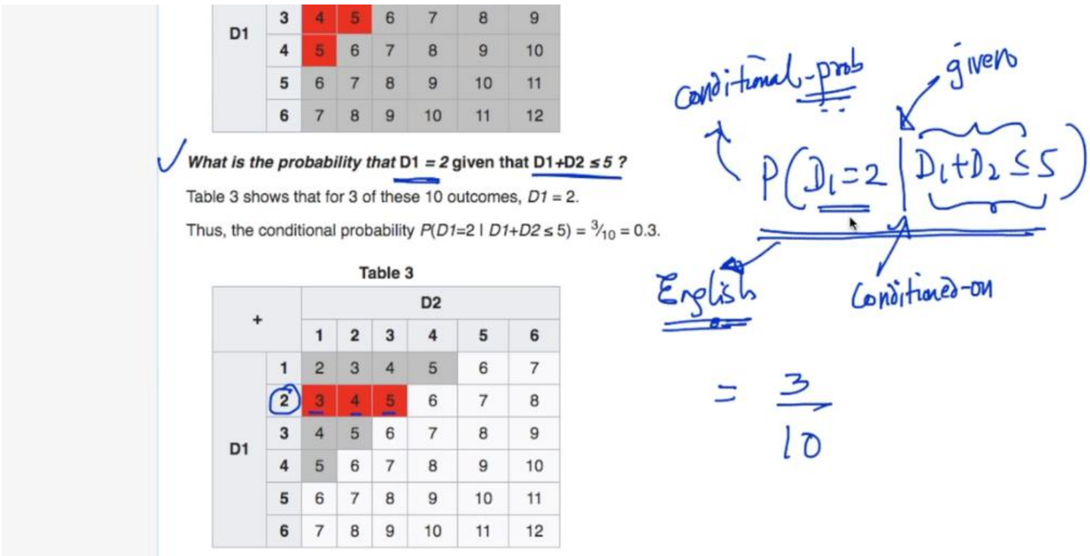
N Bayes Algorithm



D1	3	4	5	6	7	8	9
	4	5	6	7	8	9	10
	5	6	7	8	9	10	11
	6	7	8	9	10	11	12

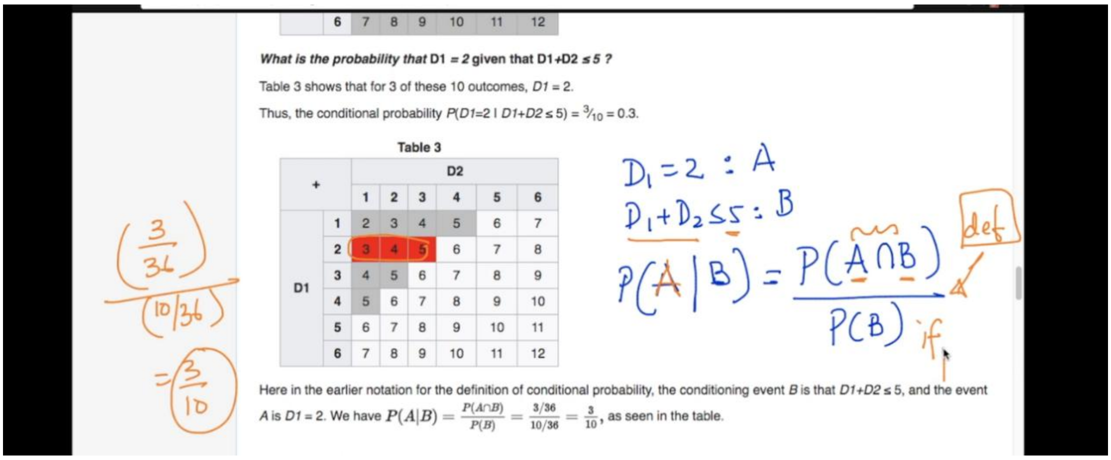
What is the probability that* D1 = *2* given that D1+D2 *≤ 5 ?

Table 3 shows that for 3 of these 10 outcomes, *D1* = 2.

Thus, the conditional probability *P*(*D1*=2 ∣ *D1*+*D2* ≤ 5) = $^3/_{10}$ = 0.3.

Table 3

+		D2					
		1	2	3	4	5	6
D1	1	2	3	4	5	6	7
	2	3	4	5	6	7	8
	3	4	5	6	7	8	9
	4	5	6	7	8	9	10
	5	6	7	8	9	10	11
	6	7	8	9	10	11	12



	6	7	8	9	10	11	12

What is the probability that* D1 = *2* given that D1+D2 *≤ 5 ?

Table 3 shows that for 3 of these 10 outcomes, *D1* = 2.

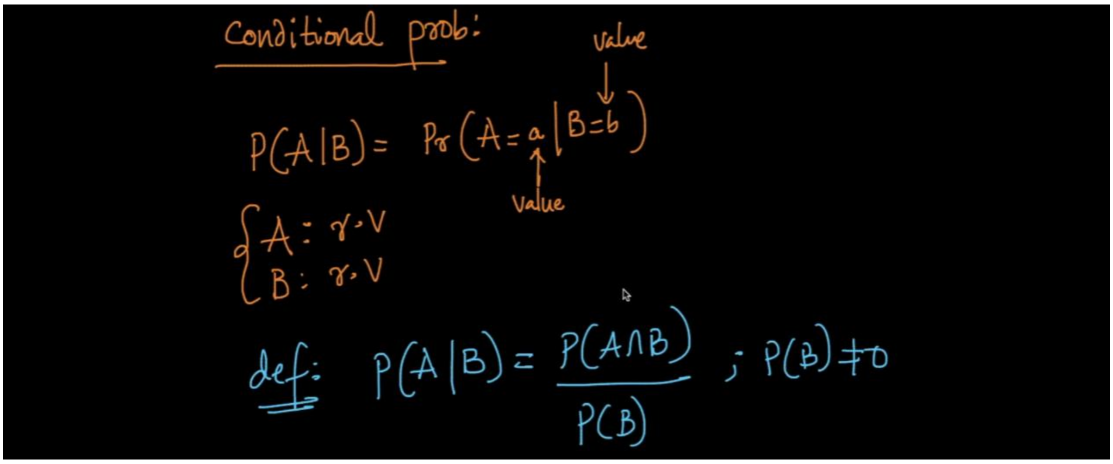
Thus, the conditional probability *P*(*D1*=2 ∣ *D1*+*D2* ≤ 5) = $^3/_{10}$ = 0.3.

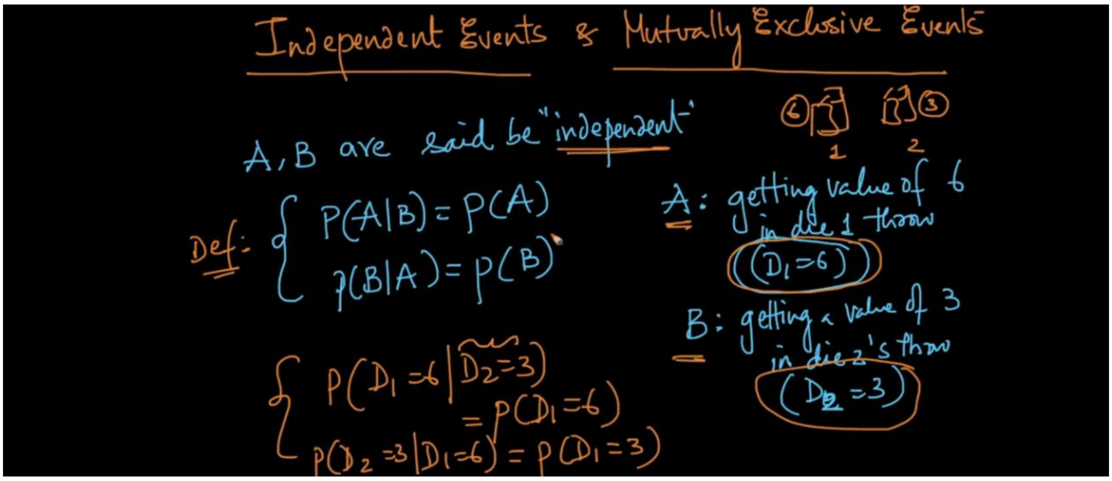
Table 3

+		D2					
		1	2	3	4	5	6
D1	1	2	3	4	5	6	7
	2	3	4	5	6	7	8
	3	4	5	6	7	8	9
	4	5	6	7	8	9	10
	5	6	7	8	9	10	11
	6	7	8	9	10	11	12

Here in the earlier notation for the definition of conditional probability, the conditioning event *B* is that *D1*+*D2* ≤ 5, and the event *A* is *D1* = 2. We have $P(A|B) = \frac{P(A\cap B)}{P(B)} = \frac{3/36}{10/36} = \frac{3}{10}$, as seen in the table.

A intersection B means what is probability of event a and b both occurs .





Now see two dice d1 and d2 . See above image p of a given b = p of a

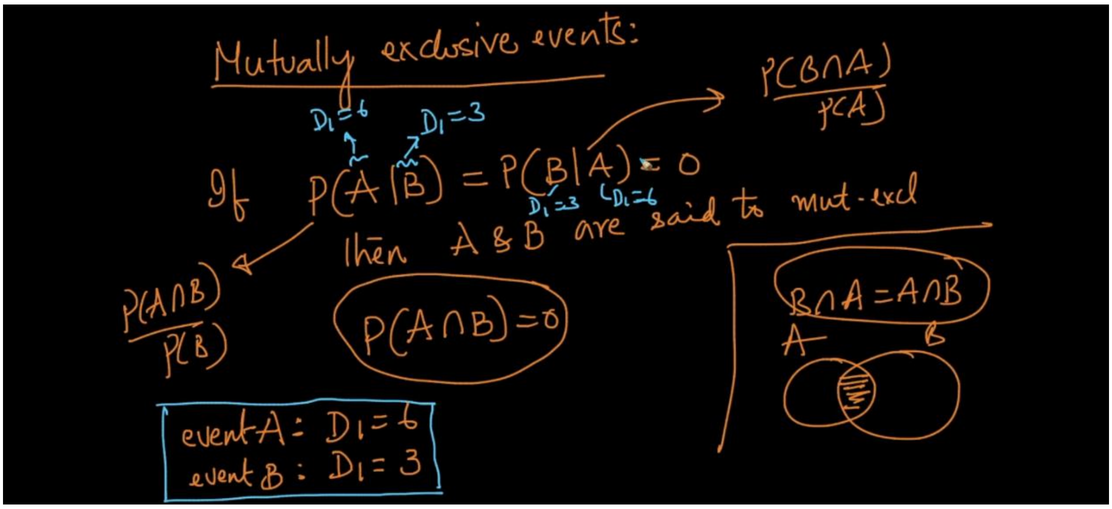
P(B|A) = P(b)

There is no relation between both dice so prob of both of them will different means not depend on each other that is called Independent events .

Mutually Exclusive :

Here see below image A is dice 1 roll p will 6
B = same dice roll probability 3 .

Now in this case dice is only one so probability is dependant now
P(A|B) = 0 (What is prob of A if B is given .)
Means here already we get P(B) = 3 so getting P(A) is zero
This condition called Mutually exclusive .



Bayes therm :

P(A|B) means prob of a and b * prob a / Prob b.

Means probability of A if b is given .

Baye's Theorem: (1700's)

Thm: $$P(A|B) = \frac{P(B|A)\,P(A)}{P(B)} \quad \text{if } P(B) \neq 0 \checkmark$$

$P(A|B)$ → posterior ✓

$P(B|A)$ → likelihood

$P(A)$ → prior

$P(B)$ → evidence

(wiki)

Proof: $$P(A|B) = \frac{P(A \cap B)}{P(B)} = \frac{P(A, B)}{P(B)}$$

A, B → $A \,\&\, B$

→ defn

$A \cap B = B \cap A$ → set theory

$$P(A|B) = \frac{P(B \cap A)}{P(B)} = \frac{P(B, A)}{P(B)} \checkmark$$

$$P(A)\,P(B|A) = \frac{P(B \cap A)}{\cancel{P(A)}} \rightarrow \text{defn.}$$

Secure | https://en.wikipedia.org/wiki/Bayes%27_theorem

denote the event that a randomly chosen item is defective. Then, we are given the following information:

$P(A_1) = 0.2, \quad P(A_2) = 0.3, \quad P(A_3) = 0.5.$

If the item was made by the first machine, then the probability that it is defective is 0.05; that is, $P(B \mid A_1) = 0.05$. Overall, we have

$P(B \mid A_1) = 0.05, \quad P(B \mid A_2) = 0.03, \quad P(B \mid A_3) = 0.01.$

To answer the original question, we first find $P(B)$. That can be done in the following way:

$P(B) = \Sigma_i P(B \mid A_i)\, P(A_i) = (0.05)(0.2) + (0.03)(0.3) + (0.01)(0.5) = 0.024.$

Hence 2.4% of the total output of the factory is defective.

We are given that B has occurred, and we want to calculate the conditional probability of A_3. By Bayes' theorem,

$P(A_3 \mid B) = P(B \mid A_3)\, P(A_3)/P(B) = (0.01)(0.50)/(0.024) = 5/24.$

Given that the item is defective, the probability that it was made by the third machine is only 5/24. Although machine 3 produces half of the total output, it produces a much smaller fraction of the defective items. Hence the knowledge that the item selected was defective enables us to replace the prior probability $P(A_3) = 1/2$ by the smaller posterior probability $P(A_3 \mid B) = 5/24$.

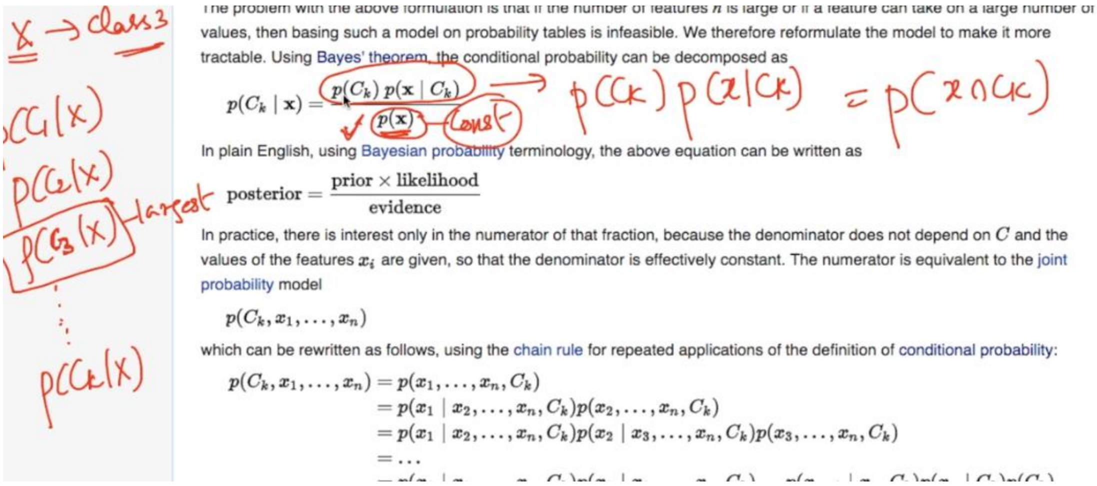
Interpretations

output

defective

$$P(A_3|B) = \frac{P(B|A_3)\,P(A_3)}{P(B)}$$

N Bayes algorithm :



The problem with the above formulation is that if the number of features n is large or if a feature can take on a large number of values, then basing such a model on probability tables is infeasible. We therefore reformulate the model to make it more tractable. Using Bayes' theorem, the conditional probability can be decomposed as

$$p(C_k \mid \mathbf{x}) = \frac{p(C_k)\ p(\mathbf{x} \mid C_k)}{p(\mathbf{x})}$$

In plain English, using Bayesian probability terminology, the above equation can be written as

$$\text{posterior} = \frac{\text{prior} \times \text{likelihood}}{\text{evidence}}$$

In practice, there is interest only in the numerator of that fraction, because the denominator does not depend on C and the values of the features x_i are given, so that the denominator is effectively constant. The numerator is equivalent to the joint probability model

$$p(C_k, x_1, \dots, x_n)$$

which can be rewritten as follows, using the chain rule for repeated applications of the definition of conditional probability:

$$\begin{aligned} p(C_k, x_1, \dots, x_n) &= p(x_1, \dots, x_n, C_k) \\ &= p(x_1 \mid x_2, \dots, x_n, C_k)\, p(x_2, \dots, x_n, C_k) \\ &= p(x_1 \mid x_2, \dots, x_n, C_k)\, p(x_2 \mid x_3, \dots, x_n, C_k)\, p(x_3, \dots, x_n, C_k) \\ &= \dots \end{aligned}$$

See above Bayes formula . C x means class from c1 ….. c n .

So here we are trying to find prob of C x given point x .

As p(X) is same for every term we will not consider it here . because we just change class c1 …. c n each time and we will pick that class whose p is very high .

So now P(C|x1…..x n) which can be written using chain rule by using definition of conditional probability .

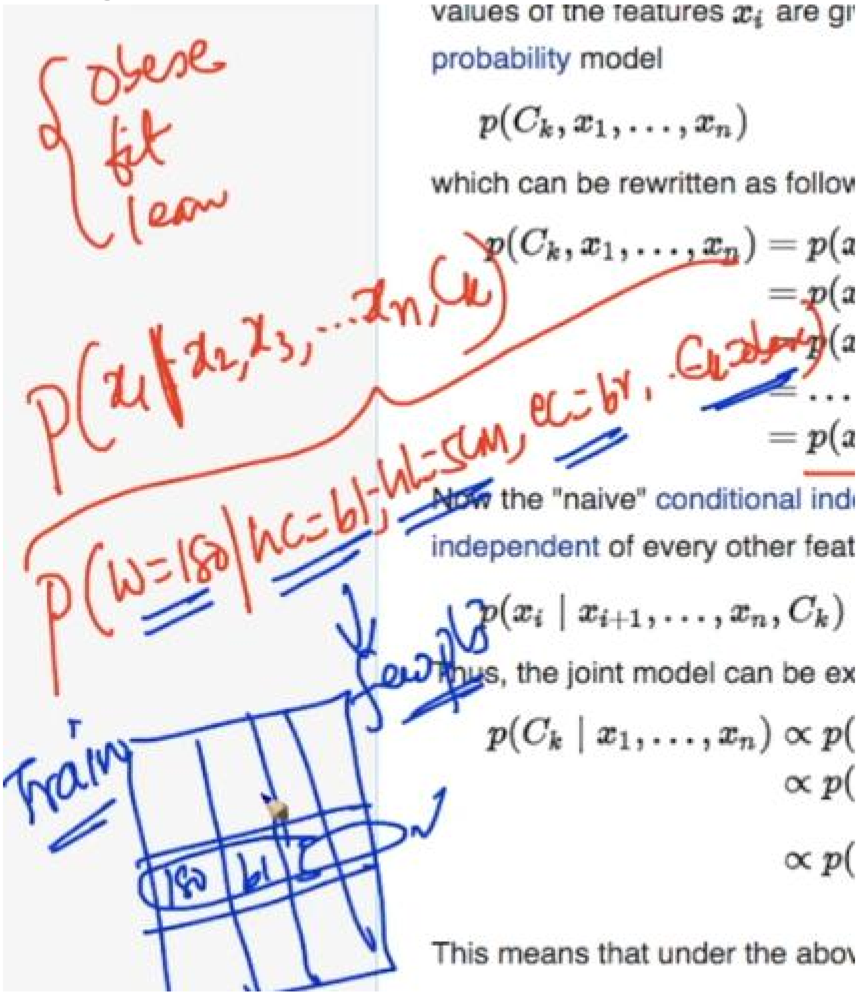
Def condition prob = P(A|B) = P(B|A)*P(B)

$p(C_k, x_1, \dots, x_n)$

which can be rewritten as follows, using the chain rule for repeated applications of the definition of conditional probability:

$$
\begin{aligned}
p(C_k, x_1, \dots, x_n) &= p(x_1, \dots, x_n, C_k) \checkmark \\
&= p(x_1 \mid x_2, \dots, x_n, C_k)\, p(x_2, \dots, x_n, C_k) \checkmark \rightarrow \text{defn. of cond. prob} \\
&= p(x_1 \mid x_2, \dots, x_n, C_k)\, p(x_2 \mid x_3, \dots, x_n, C_k)\, p(x_3, \dots, x_n, C_k) \\
&= \dots \\
&= p(x_1 \mid x_2, \dots, x_n, C_k)\, p(x_2 \mid x_3, \dots, x_n, C_k) \dots p(x_{n-1} \mid x_n, C_k)\, p(x_n \mid C_k)\, p(C_k)
\end{aligned}
$$

Now computing this probability is really very hard in big train data .



Lets see above image I want to predict P(weight = 180 | hair color = blue,length = 5cm…,class = fat)

So finding this exact data is very very low prob .

So as per N Bayes algorithm :

s means that

$$P(A|B) = P(A)$$

$$P(A|\underbrace{B,C}_{B\cap C}) = P(A|C)$$

See first a and B are simple independent because there is no rel between them .

In second we can a and B are conditionally independent . ..

So in N algorithm we Conditional ind concept .

So above image complex equation we make simple :

See below image now we are telling here x I is conditionally independent from x(I)+1 ..

$$\begin{aligned} p(C_k, x_1, \dots, x_n) &= p(x_1, \dots, x_n, C_k) \\ &= p(x_1 \mid x_2, \dots, x_n, C_k)\, p(x_2, \dots, x_n, C_k) \\ &= p(x_1 \mid x_2, \dots, x_n, C_k)\, p(x_2 \mid x_3, \dots, x_n, C_k)\, p(x_3, \dots, x_n, C_k) \\ &= \dots \\ &= p(x_1 \mid x_2, \dots, x_n, C_k)\, p(x_2 \mid x_3, \dots, x_n, C_k) \dots p(x_{n-1} \mid x_n, C_k)\, p(x_n \mid C_k)\, p(C_k) \end{aligned}$$

Now the "naive" conditional independence assumptions come into play: assume that each feature x_i is conditionally independent of every other feature x_j for $j \neq i$, given the category C. This means that

$$p(x_i \mid x_{i+1}, \dots, x_n, C_k) = p(x_i \mid C_k)\,.$$

$\Rightarrow$ x_1 is indep (cond) of x_{i+1}, x_{i+2} .... x_n given C_k

Thus, the joint model can be expressed as

$$p(x_i \mid x_{i+1}, \dots, x_n, C_k) = p(x_i \mid C_k)\,.$$

Thus, the joint model can be expressed as

$$\begin{aligned} p(C_k \mid x_1, \dots, x_n) &\propto p(C_k, x_1, \dots, x_n) \\ &\propto p(C_k)\, p(x_1 \mid C_k)\, p(x_2 \mid C_k)\, p(x_3 \mid C_k) \cdots \\ &\propto p(C_k) \prod_{i=1}^{n} p(x_i \mid C_k)\,. \end{aligned}$$

This means that under the above independence assumptions, the conditional distribution over the class variable C is:

$$p(C_k \mid x_1, \dots, x_n) = \frac{1}{Z} p(C_k) \prod_{i=1}^{n} p(x_i \mid C_k)$$

where the evidence $Z = p(\mathbf{x}) = \sum_k p(C_k)\, p(\mathbf{x} \mid C_k)$ is a scaling factor dependent only on $x_1, \dots, x_n$, that is, a constant if the values of the feature variables are known.

Pi is used to denote multiplication operation .

Z = p(X) which we neglect on top .

Contents - Google Docs | ShatterLine Blog » Not-so-N... | Naive Bayes classifier - Wiki...

shatterline.com/blog/2013/09/12/not-so-naive-classification-with-the-naive-bayes-classifier/

The Learning Phase

In the learning phase, we compute the table of likelihoods (probabilities) from the training data. They are:

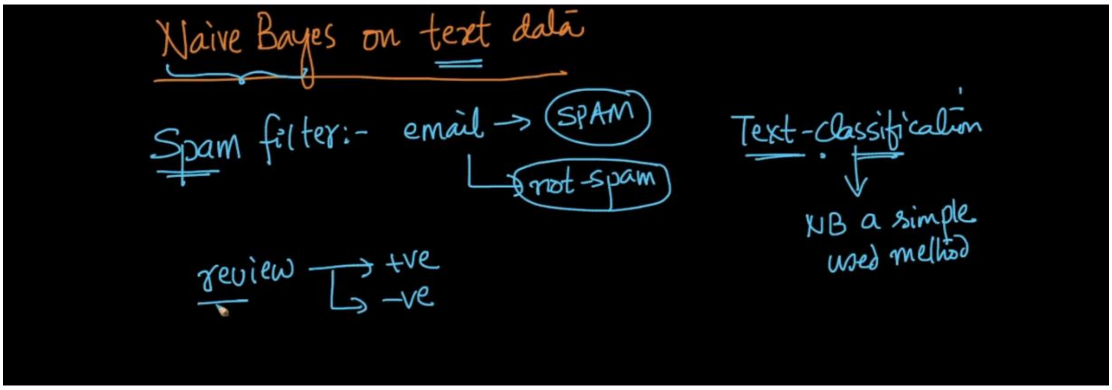
P(Outlook=o|Class$_{Play=b}$), where o ∈ [Sunny, Overcast, Rainy] and b ∈ [yes, no]

P(Temperature=t|Class$_{Play=b}$), where t ∈ [Hot, Mild, Cool] and b ∈ [yes, no],

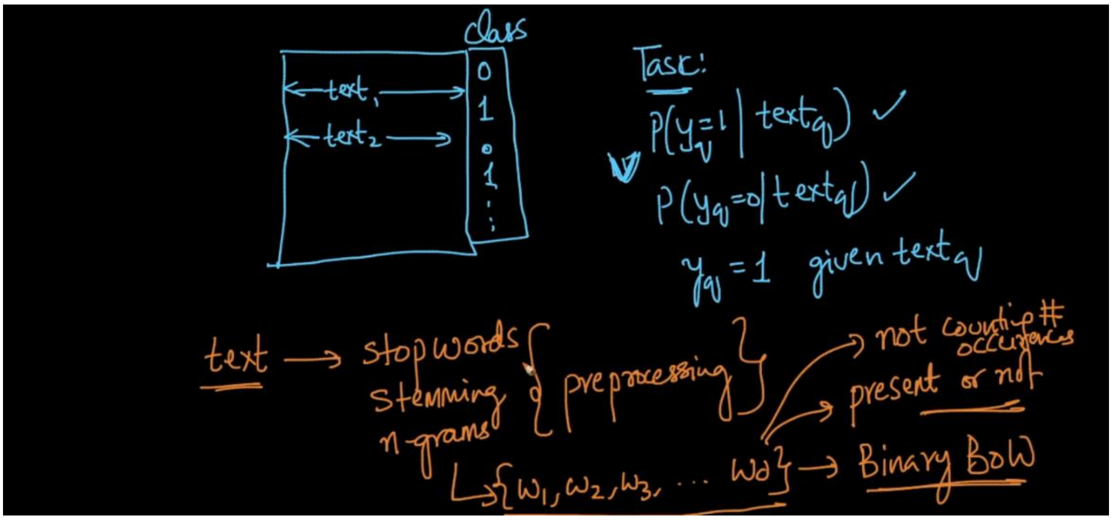
P(Humidity=h|Class$_{Play=b}$), where h∈ [High, Norma] and b ∈ [yes, no],

P(Wind=w|Class$_{Play=b}$), where w ∈ [Weak, Strong] and b ∈ [yes, no].

P(Outlook=o|Class $_{Play=Yes\|No}$)	Frequency		Probability in Class	
Outlook =	Play=Yes	Play=No	Play=Yes	Play=No
Sunny	2	3	2/9	3/5
Overcast	4	0	4/9	0/5
Rain	3	2	3/9	2/5
	total= 9	total=5		



Now our task is text classification here we have text data with 2 class label 0 and 1 we need to predict new text class .



Using N Bayes we can write :

P(y=1 | Text) = p(y=1 | w1,w2,……. w n)

Simply we can use N Bayes formula . :

$$\underline{\text{text}} \xrightarrow{\text{prepro}} \underline{\{w_1, w_2, \dots, w_d\}}$$

$$P(y=1 \mid \text{text}) = P(y=1 \mid \underbrace{w_1, w_2, \dots, w_d}_{\text{features}})$$

class prior

$$\propto P(y=1) * P(w_1 \mid y=1) * P(w_2 \mid y=1) \dots P(w_d \mid y=1)$$

likelihood

$$\propto P(y=1) * \prod_{i=1}^{d} P(w_i \mid y=1)$$

$$\underline{P(y=0 \mid \text{text})} \propto P(y=0) * \prod_{i=1}^{d} \underline{P(w_i \mid y=0)}$$

$$P(y=1) = \frac{\#\ \text{Train pts with } y=1}{\text{Total } \#\ \text{Train pts}}$$

$$P(y=0) = \frac{\#\ \text{train pts with } y=0}{\text{Total } \#\ \text{train pts}}$$

← text_i →

0
1
0
0
1

Train

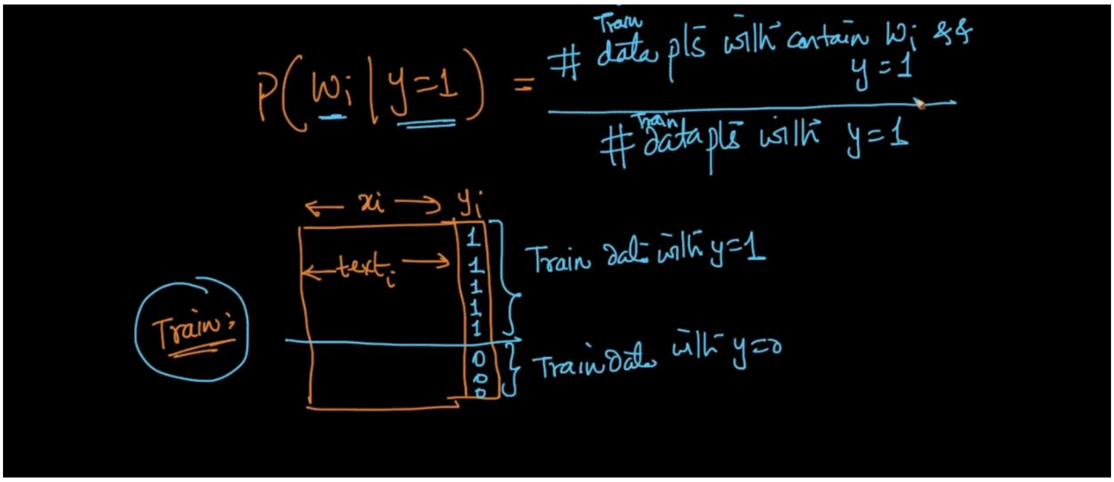
P(Y=1) and P(Y=0) we can easily cal but main task is likelihood prob find .
How can we do this ?

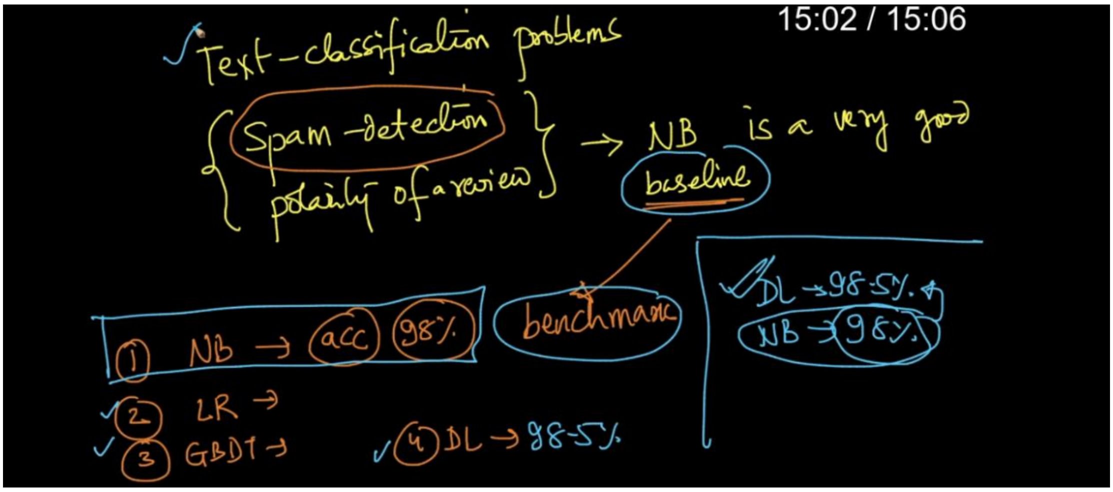
Simple we split train data half part like all class label 1 point separate from 0 points .

Simply then we can find probability for each word in text .

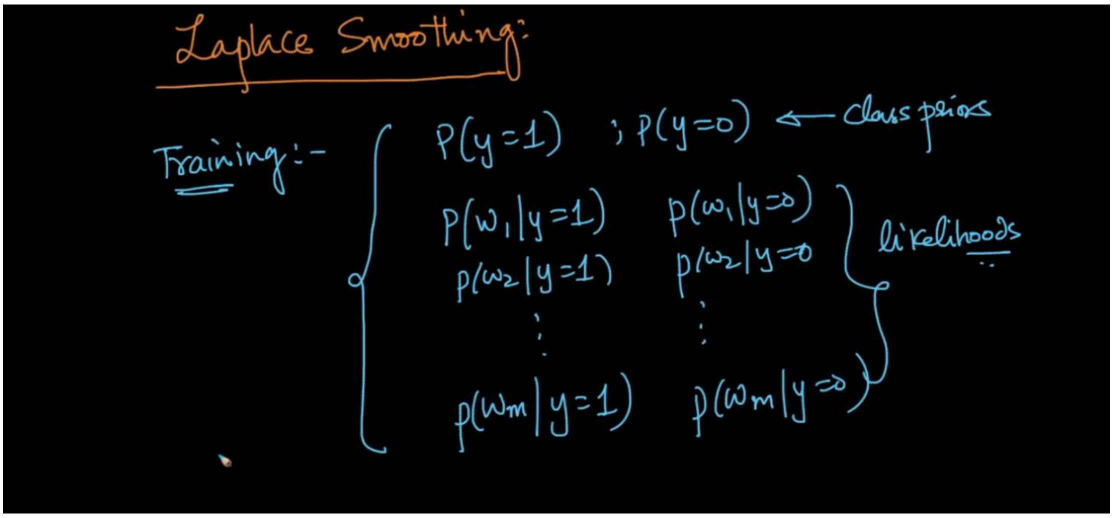
See below image in this way we can find P for all words in a text and we can tell whether new text belongs to class 1 or 0 .

Text classification performance By N algorithm is very very simple and very good also .





Laplace Smoothing :

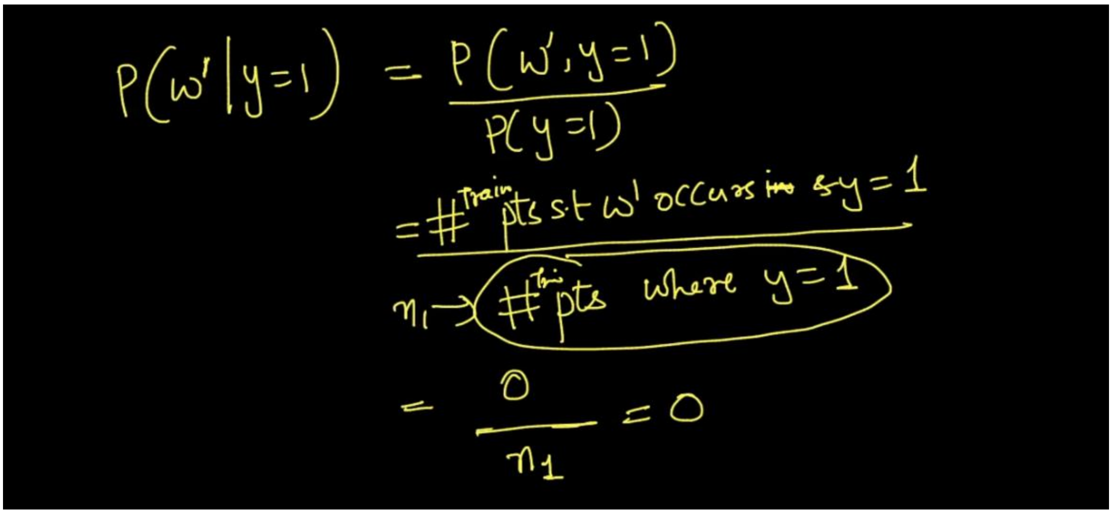


While training all P are computed for all words .

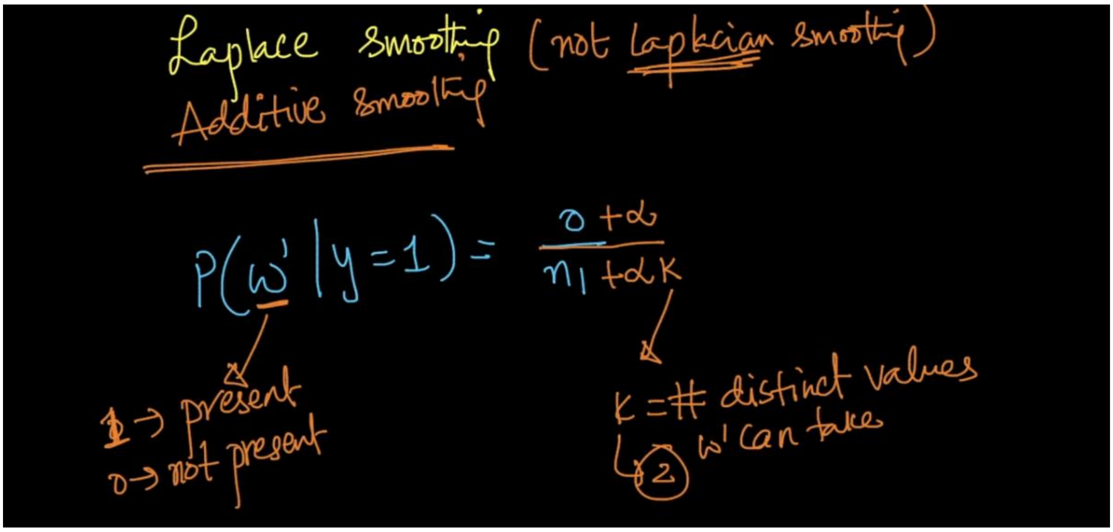
Now while testing new word come and its not available in train then how we will solve this problem .

So we cant directly ignore that word we are ignoring any word means indirectly we are telling that new word belong to class 1 . this is wrong .

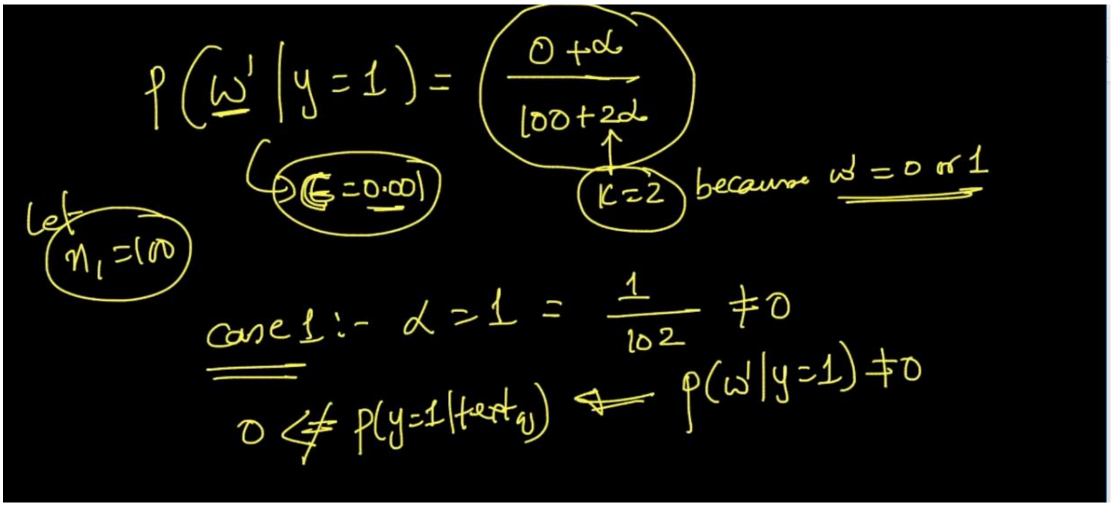
See below image



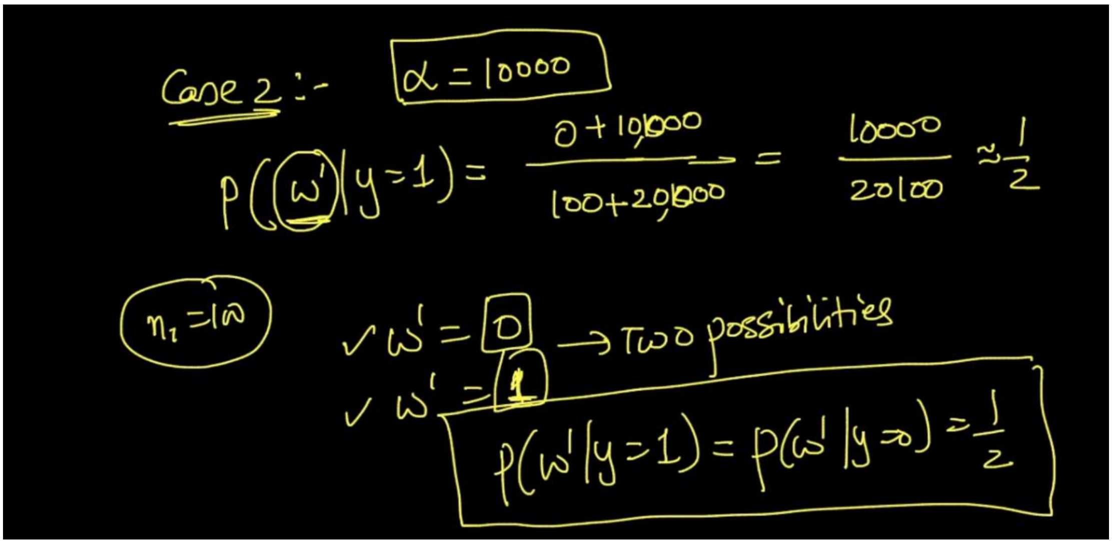
See instead of 0 we add alpha see below image and k means distinct possibility in this case only 2 hence k =2 .   means word is present or not present only 2 possibility .



How solve that 0 problem in image 1 . simple lets say n = 100 here and alpha =1 if we put this value we will get solution from that zero problem .

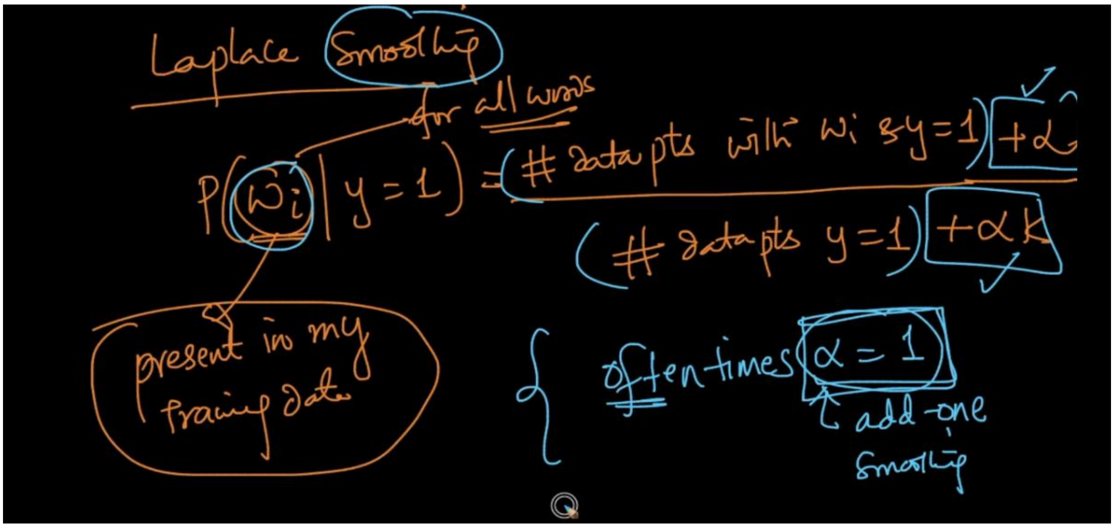


Lets take 2nd case alpha = 10k
So here we tell that p(w’|y=1)=p(w’|y=0) is equally same .
Ans this is too good because we don’t know anything about that new word .



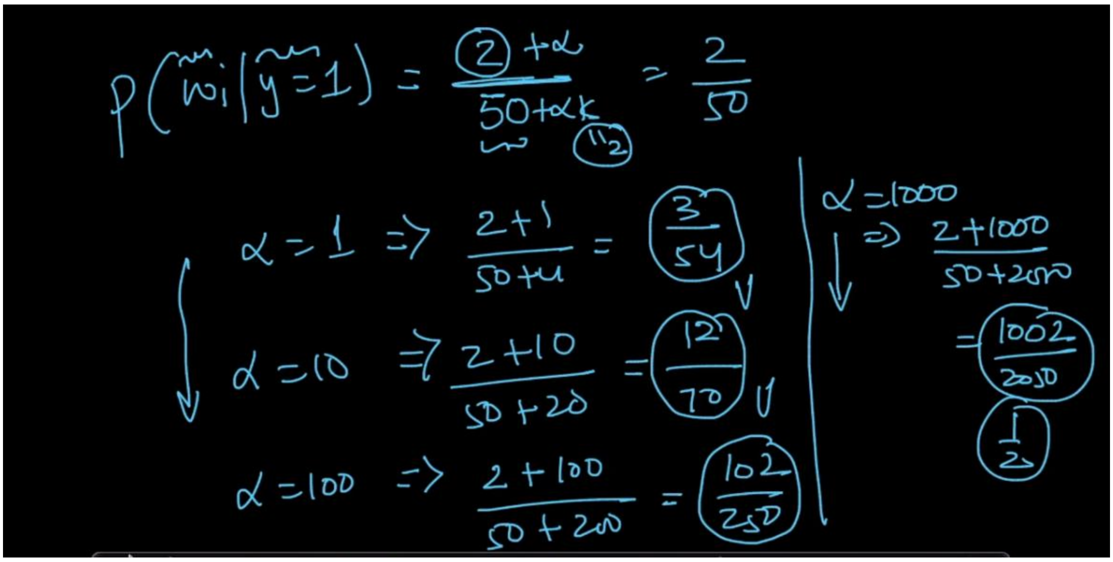
In this way Laplace will help to solve the problem of unknown word .

In real what we can do we take Laplace for all points means even word is present or not we add Laplace term see below images .



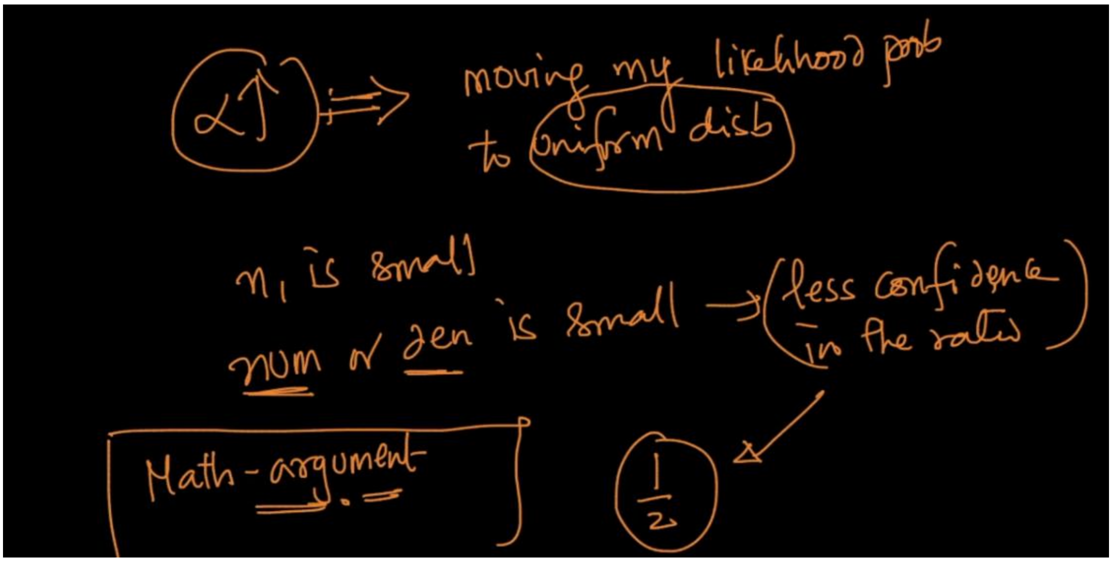
Default value is alpha =1 as we increase alpha we move towards uniform dist .

Lets see below image say we have very less data , means word w1 occurs only 2 times and P of y = 1 only 50 points are in my data set .

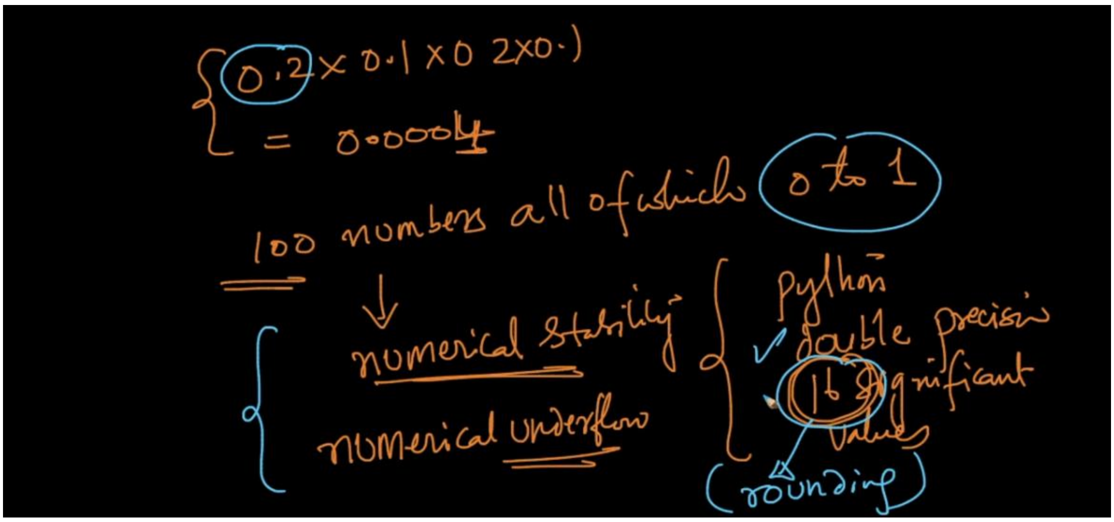


See as alpha increase we are moving toward uniform dist . because at the end we get almost 1/2 value .

Ans this is really good because if we see above example confidence on data is very low so its better to predict 50% (instead of say 1 or 0) ..



Log Probabilities :



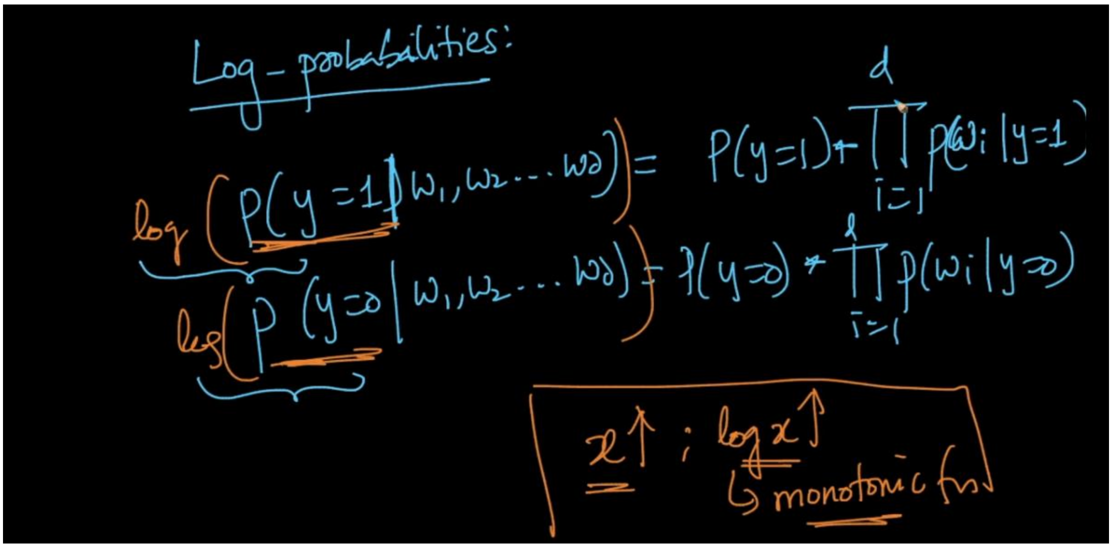
We know that P values always between 0 to 1 so its very difficult to maintain number stability .

Lets p = 0.0004 .

Also multiplying small numbers is really very hard so we convert values of P in log ..

Log has very beautiful properties :

1. Convert multiplication into sum .
2. Convert exp into multiplication .



Above N Bayes formula also convert to sum .

$$\log\left(P(y=1 \mid w_1, w_2 \dots w_d)\right)$$

$$= \log\left(P(y=1)\right) + \sum_{i=1}^{d} \log\left(P(w_i \mid y=1)\right)$$

$$\log(\underbrace{a \cdot b}_{\text{mult}}) = \underbrace{\log a + \log b}_{\text{addn}}$$

So if values are very small then log prob will be big negative number so sum of it is really very easy and time saving also .

Bias Variance Trade Off N Bayes Laplace :

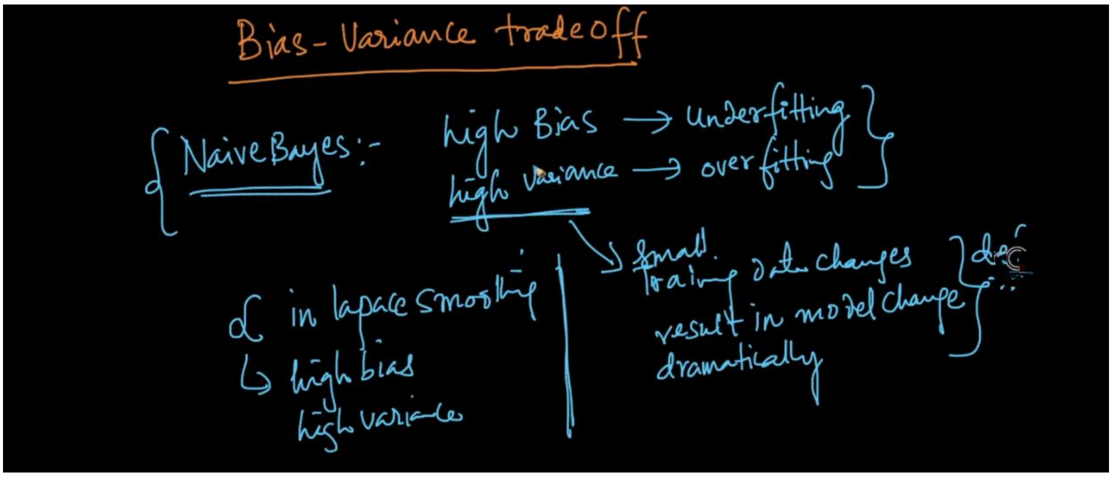
High bias = Under fir
High Variance = Over fit.

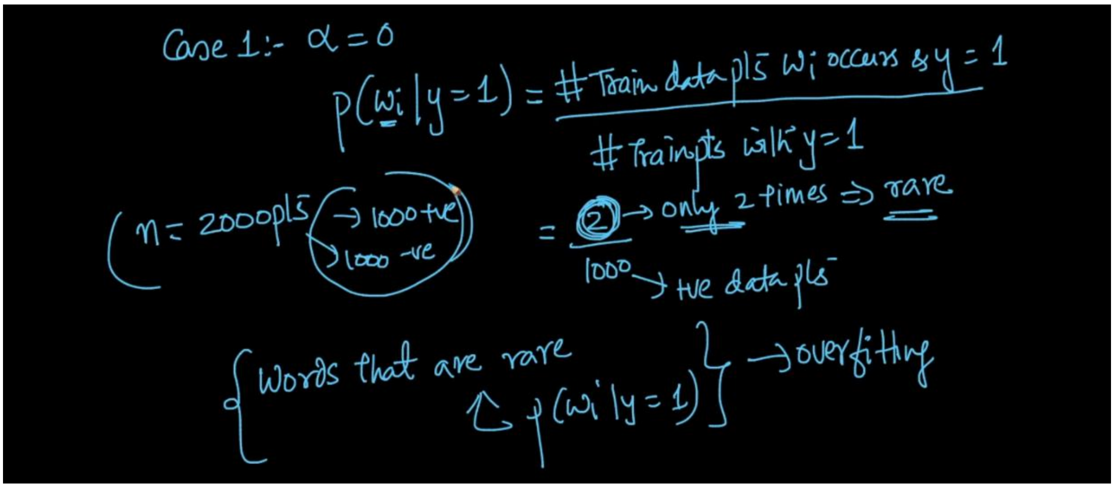
Here this is depend on value of alpha . same like KNN when k = 1 very small we saw over fit model and if K = n then under fit model .

Now lets take Case 1:
Alpha = 0 .
Now lets say we have data set of 2000 values out of that 1000 are positive and 1000 are neg points.

Now new word come and we want to predict whether it belong to + or - .

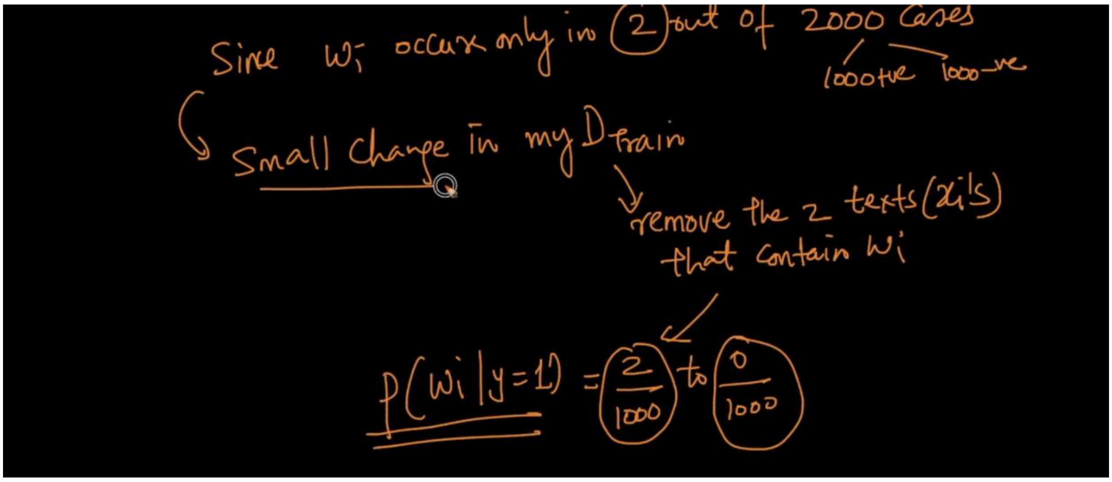
We saw in our data that word occurs very very low :
2/1000 2 times only .

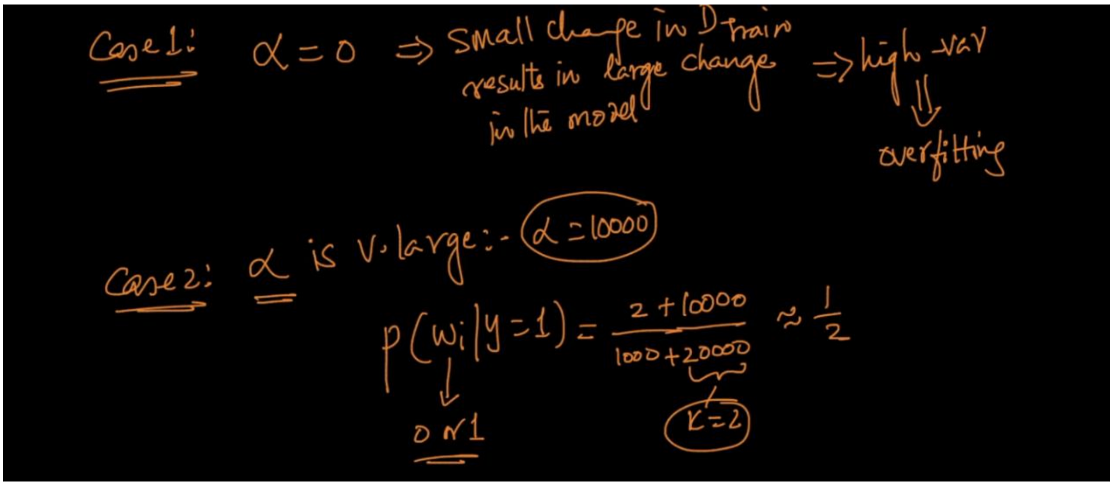


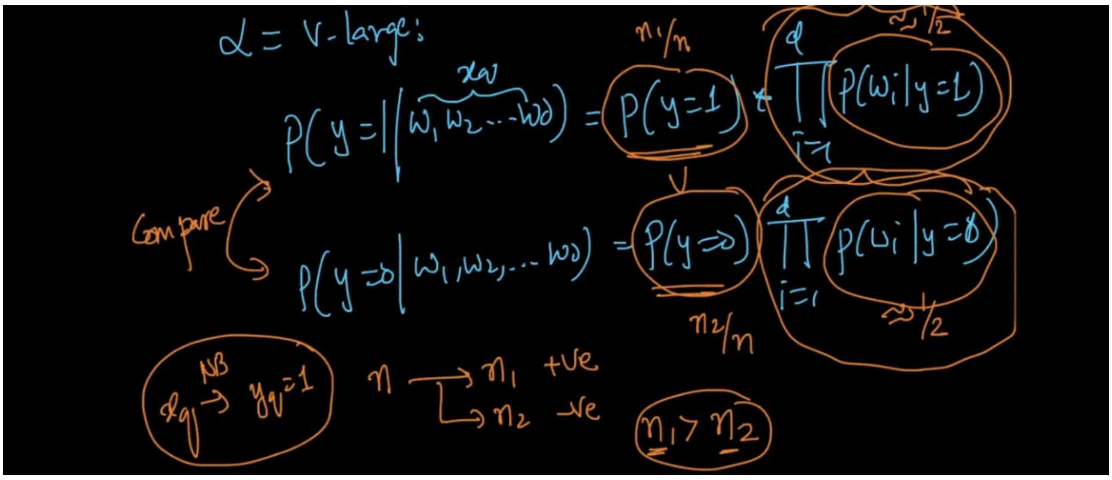


So as that word occurs only 2 times if I made very small change in data means I just remove that word then I will get 0/1000 means direct 0 prob .

See how drastic change here occurs and this is called high variance over fit problem .

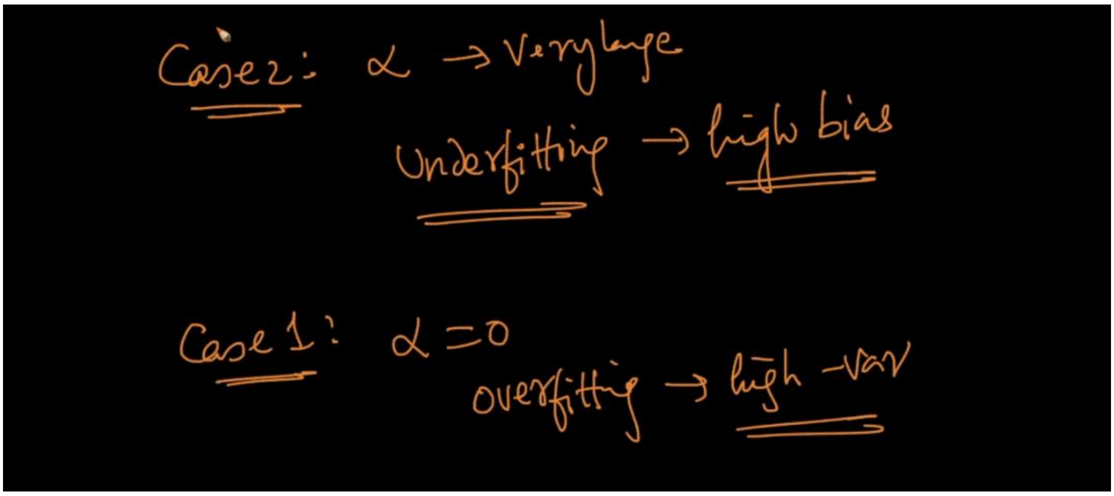


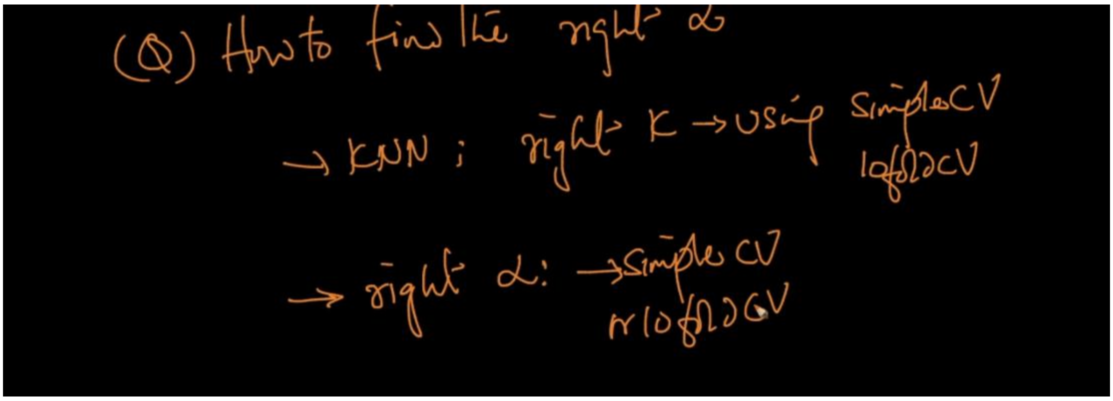


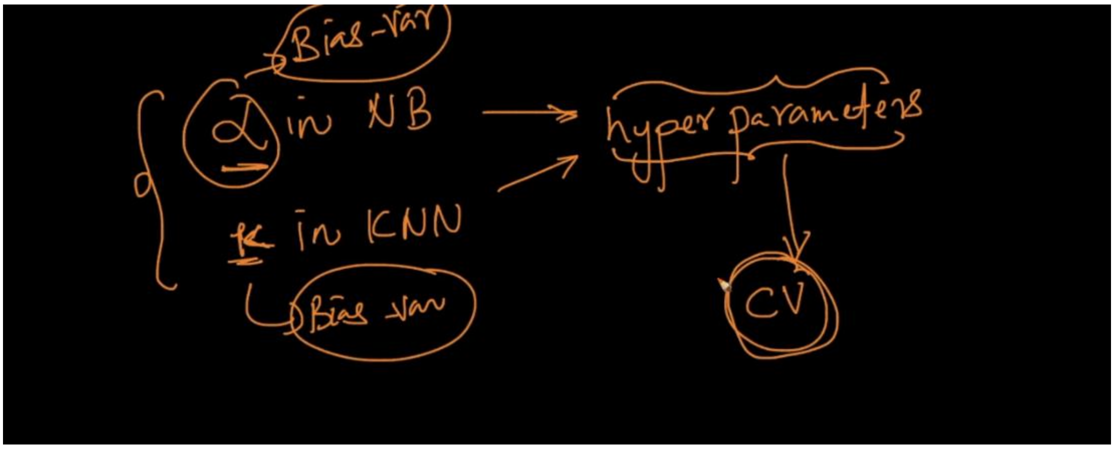


Now if alpha is very large then It will behave same like KNN whatever class has high probability then it will blindly declare new point belongs to that class only .

This is high bias problem .



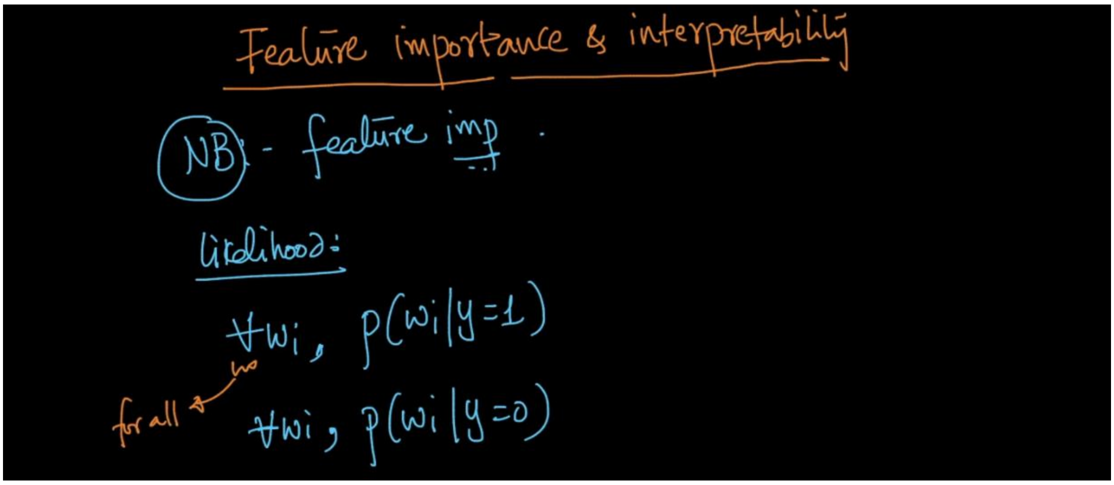


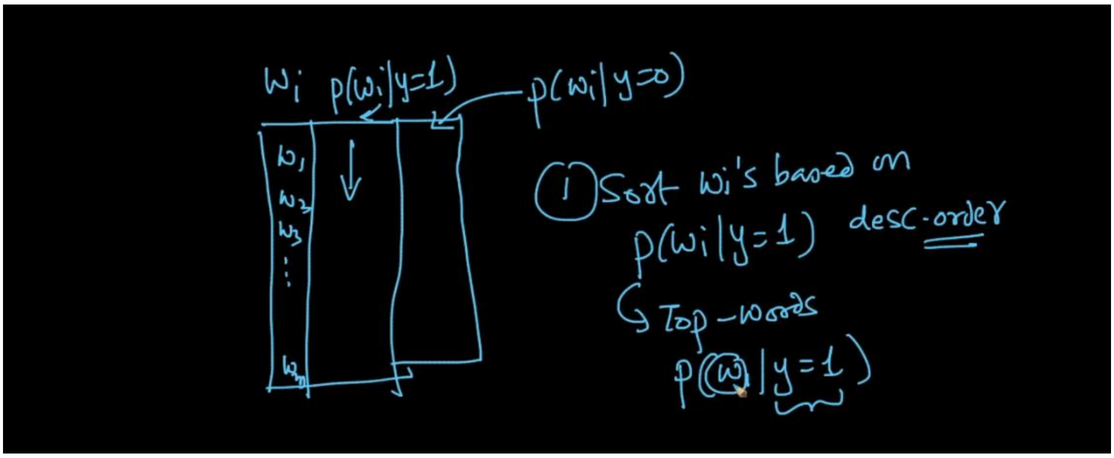


Alpha and K are hyper parameter and they cal by Cross validation method .

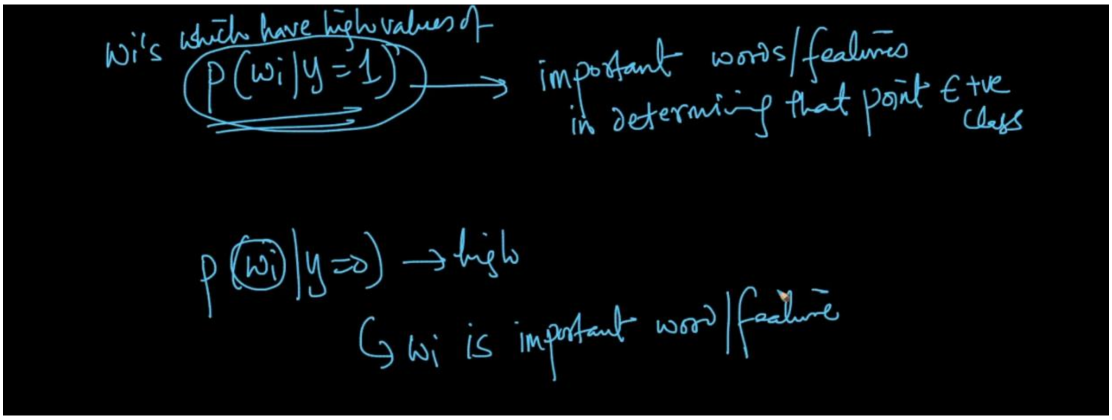
Feature Importance :

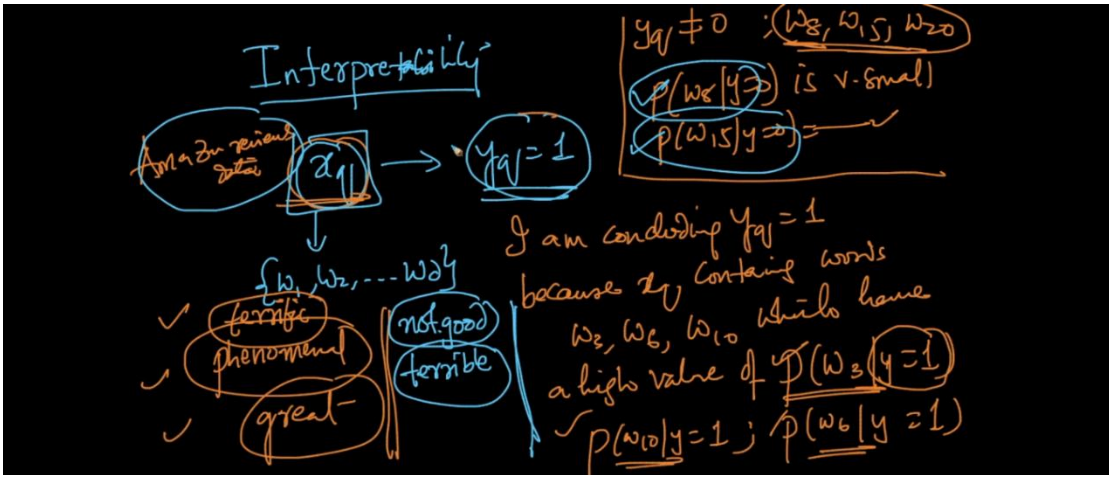
Here while training model we have P of all words . So just by sorting them we can tell which feature is important .





See below image lets say we have w1 feature if its likelihood P is very with then we can say its imp feature belong to class 1 or 0 .



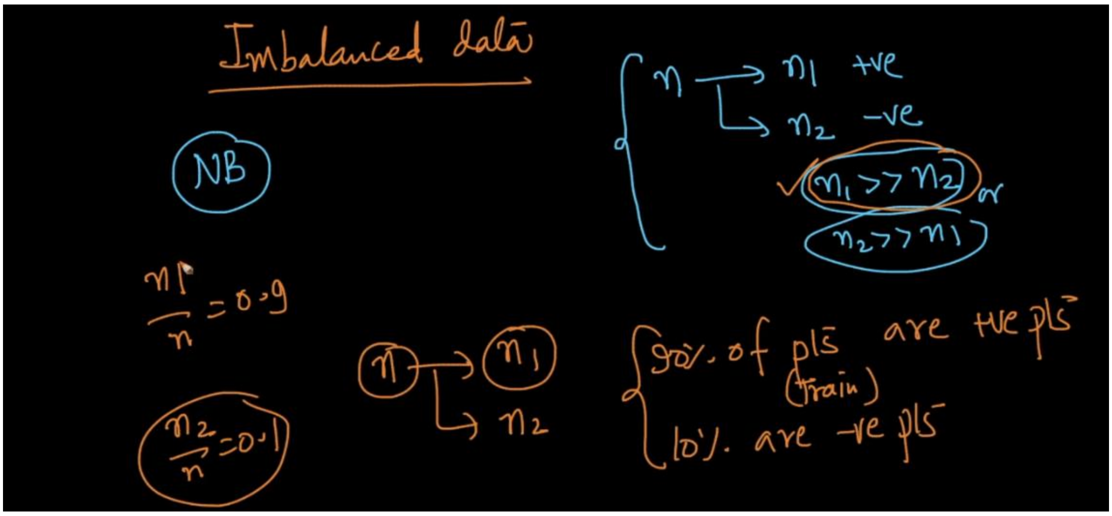


Now how interpret N Bayes is ?

Its really very Interpret lets say new review come and we want to predict review is + or - and also why ?

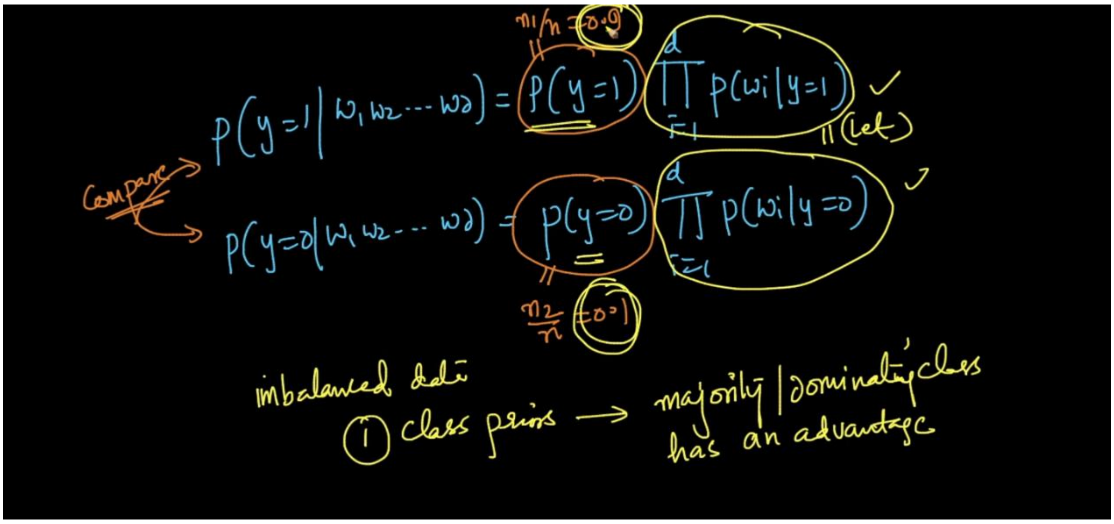
Simple our model will break review into words and we already cal prob of all train data words so it will check that prob for each word in new sent and as per that prob it will say I found w1,w3,w4 in new review has very high + prob and for w2,w7 … low probability (-) so this review is + ..

Imbalance Data :



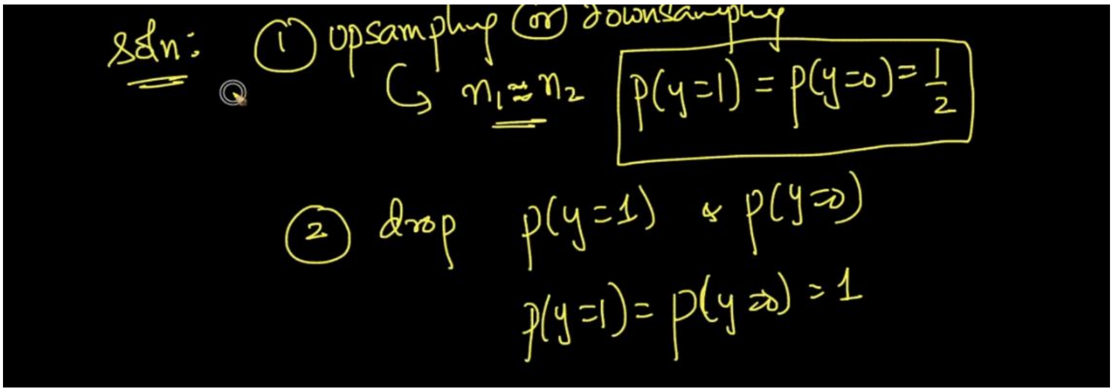
See above image data is totally imbalanced n1 too much greater than n2 .

At the we have to compare all the prob that is our obj .



See above image 1st value is 9 times greater than second so it will affect too much on model output .

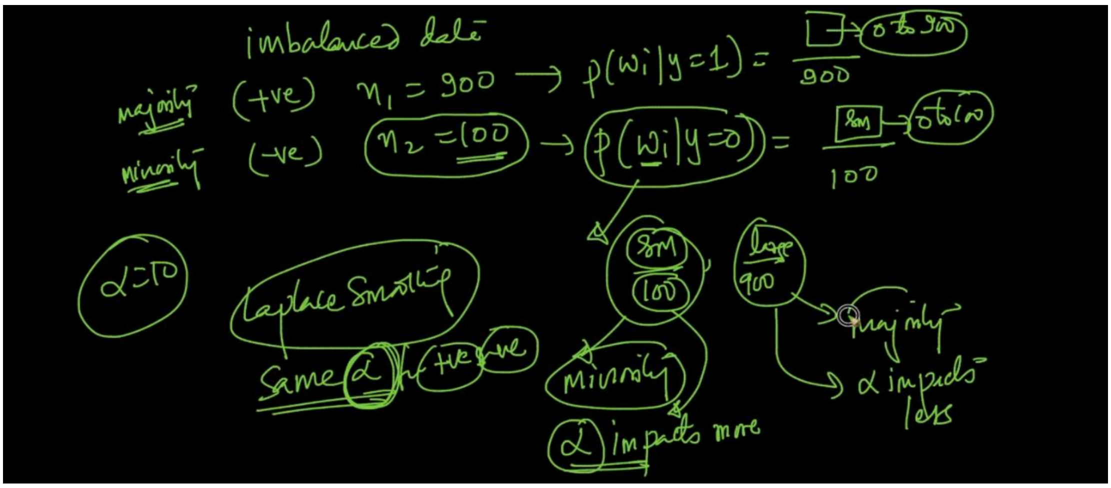
Because for any single word we get high probability in 1st case than 2nd .



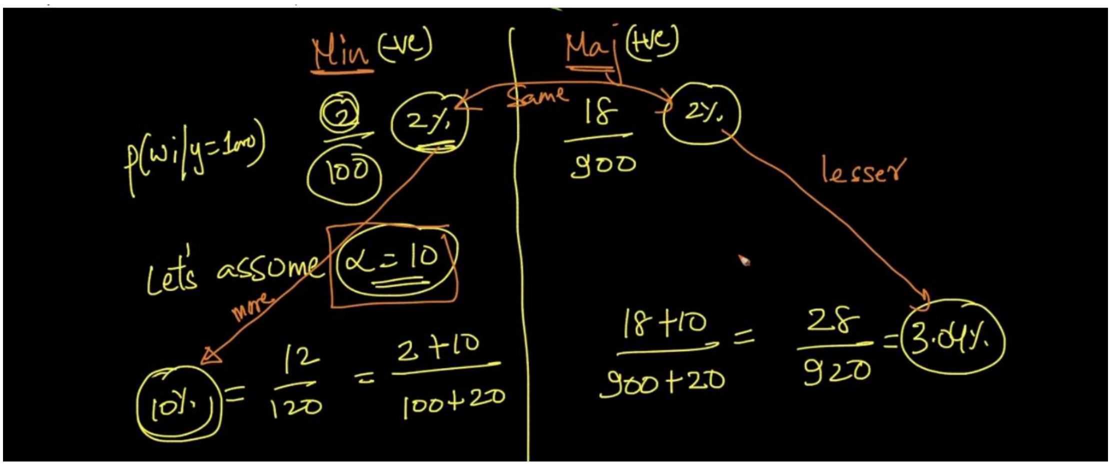
We use above we can make n1 = n2 or n2 = n1 ..

Another problem with not balanced data :

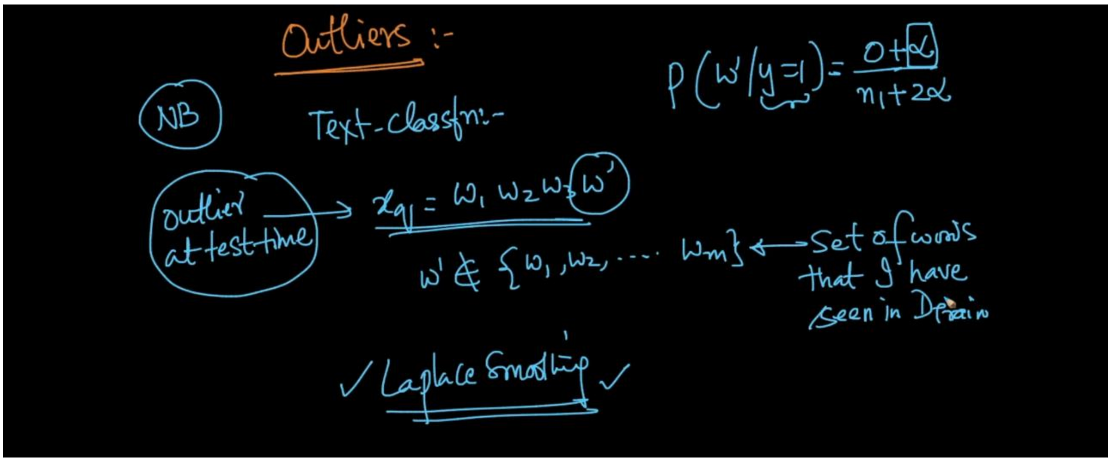
Lets say we are using Laplace then impact of alpha on minority class will be very high and very low on majority class .



See below image we start for both class with 2 % data and at the end we get very diff result minority class with high prob as compare to majority class.



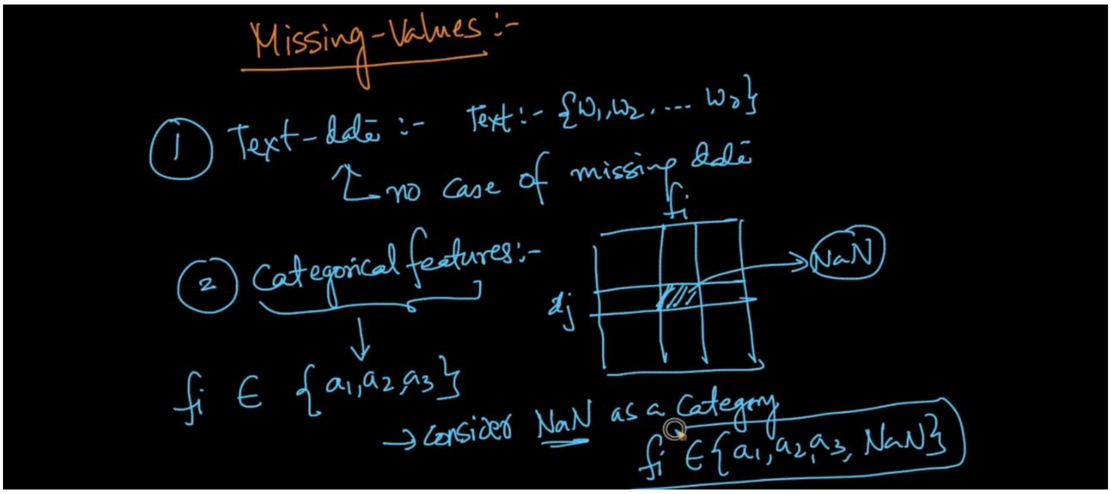
Out Li :



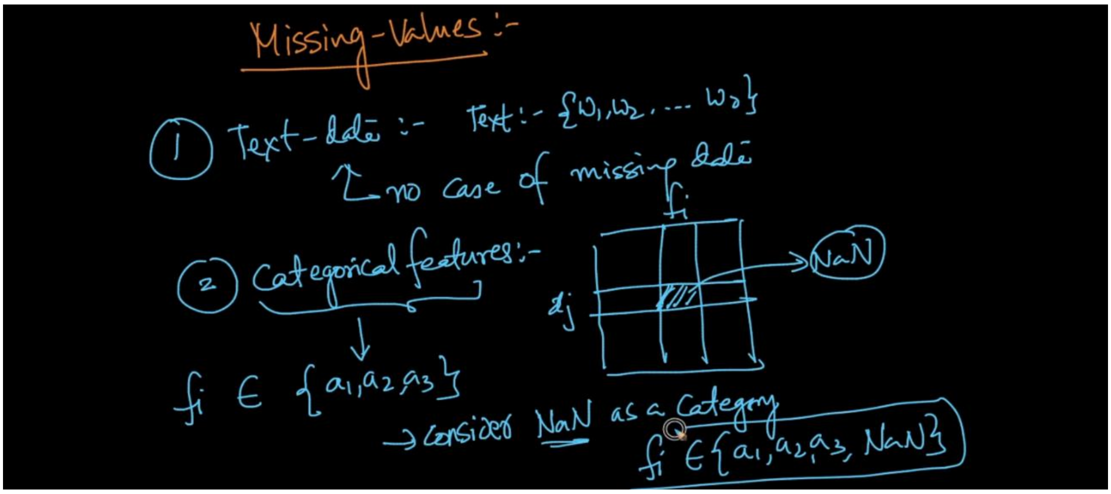
If any word occurs very very less time on both class in then it is out l.

We can avoid simply by telling if any word lets say occurs less than 10 then just ignore that word.

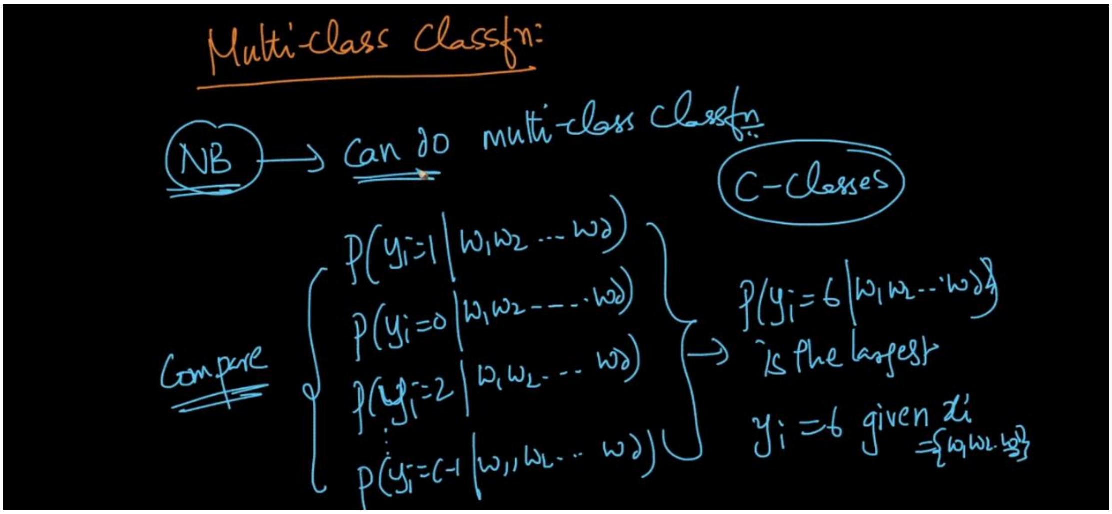
Another method is simply use Laplace Smoothie with good alpha .

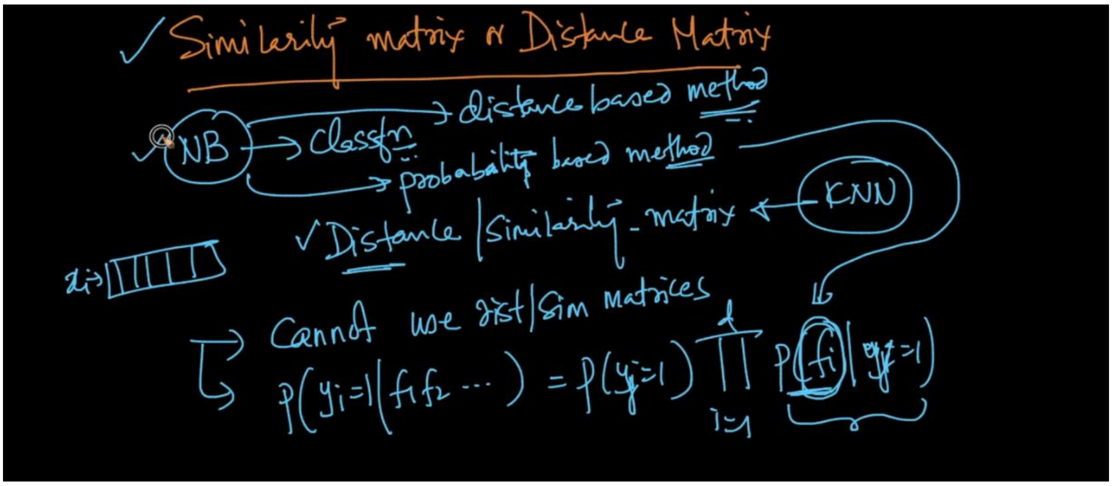


For text data no missing values because text is group of word w1,w2,w3 … so no missing .



For 2nd type whatever missing with Nan values we will consider it self as new category .





Because N Bayes is not distance based method it is probabilistic method.

N Bayes can easily used for very large dimension data just make sure use log of prob .

Large dimensionality

NB → Text Classification → high-dim

↳ log-probabilities ✓ → numerical stability

$$P(y=1 \mid w_1, w_2 \ldots w_d) = P(y=1)\prod_{i=1}^{d} P(w_i \mid y=1)$$

d → d is large

-low to

$0 < x < 1$

---

(2) Text-classfn → email spam
→ review polarity {high-dim}

↓ NB ✓ → baseline | benchmark

(3) Categorical features ✓

✓ real-valued features → seldom NB

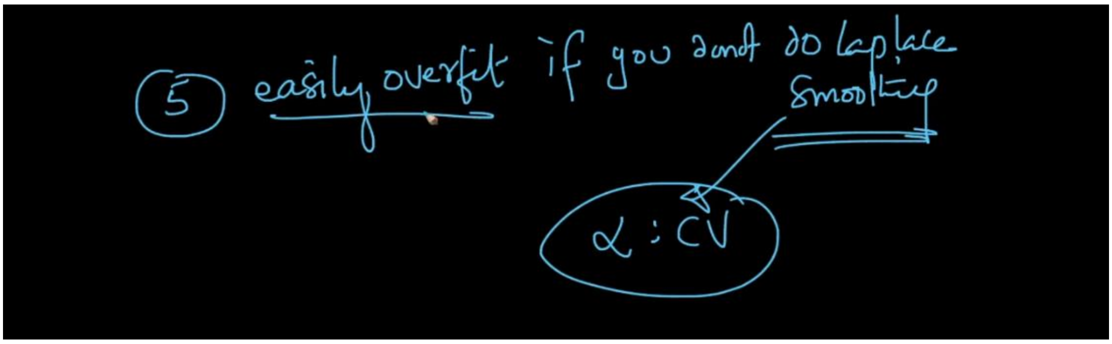
Gaussian NB / power-law ✓

---

(4) Interpretable, feat-importance → medical

✓ runtime complx → low
train time comple → low
run time space → low (prior, likelihood)

↳ NB : counting → simple
→ simple to implement



Learn code from SKLEARN library .