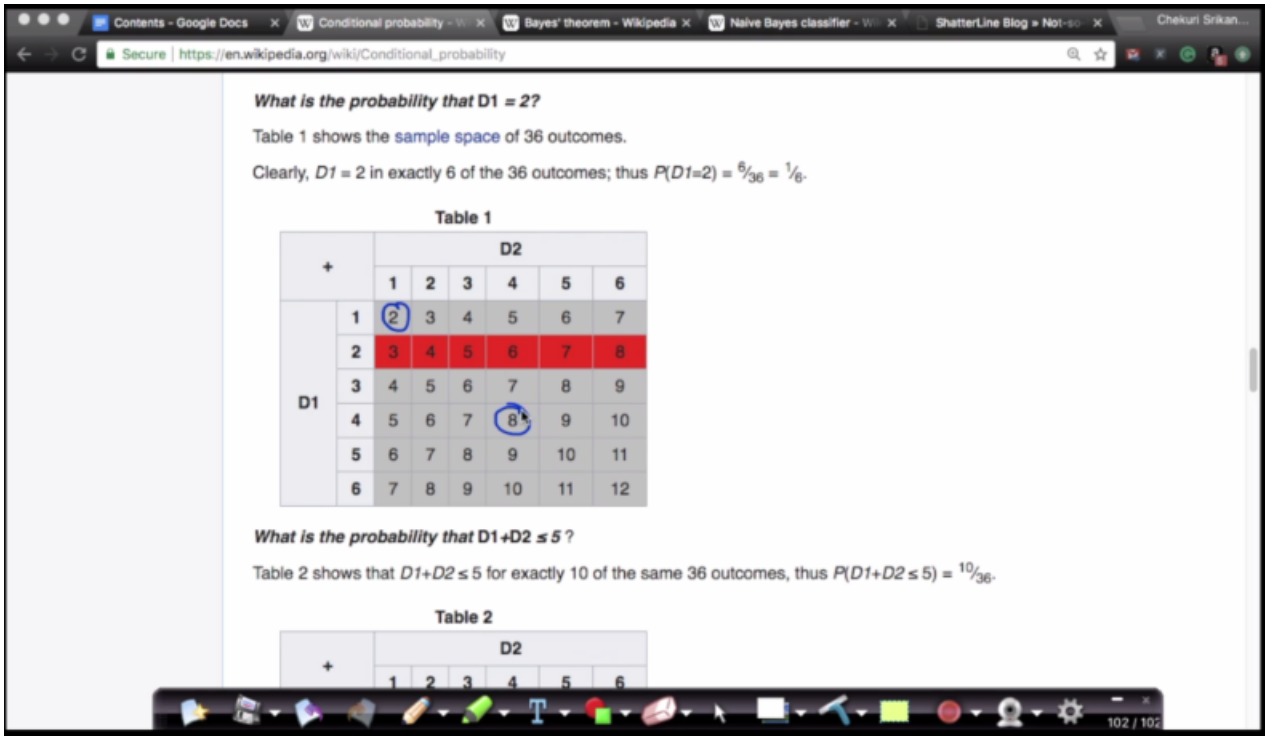
NOTES

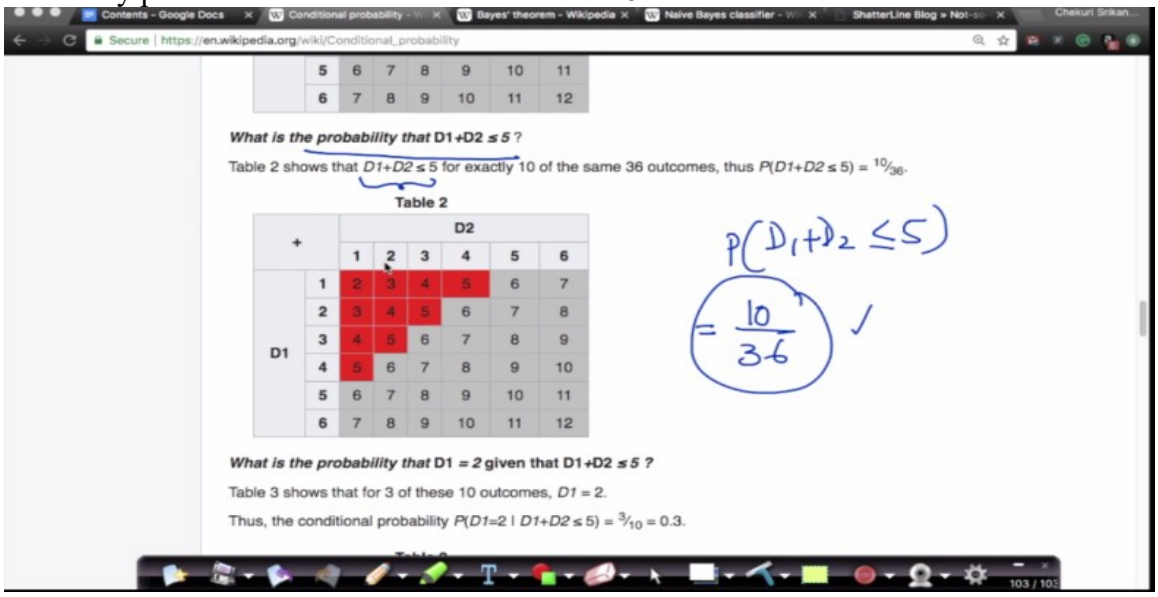
Naive Bayes is the classification algorithm. I is  based on the probability.
Conditional Probability:
It is the probability of A given B. Where A and B are the two random variables.

Example:

The outcome of one die is 6 possibilities and the outcome of other die is 6 possibilities. When we sum the outcomes of the two dies then there is a sample space of 36 outcomes.

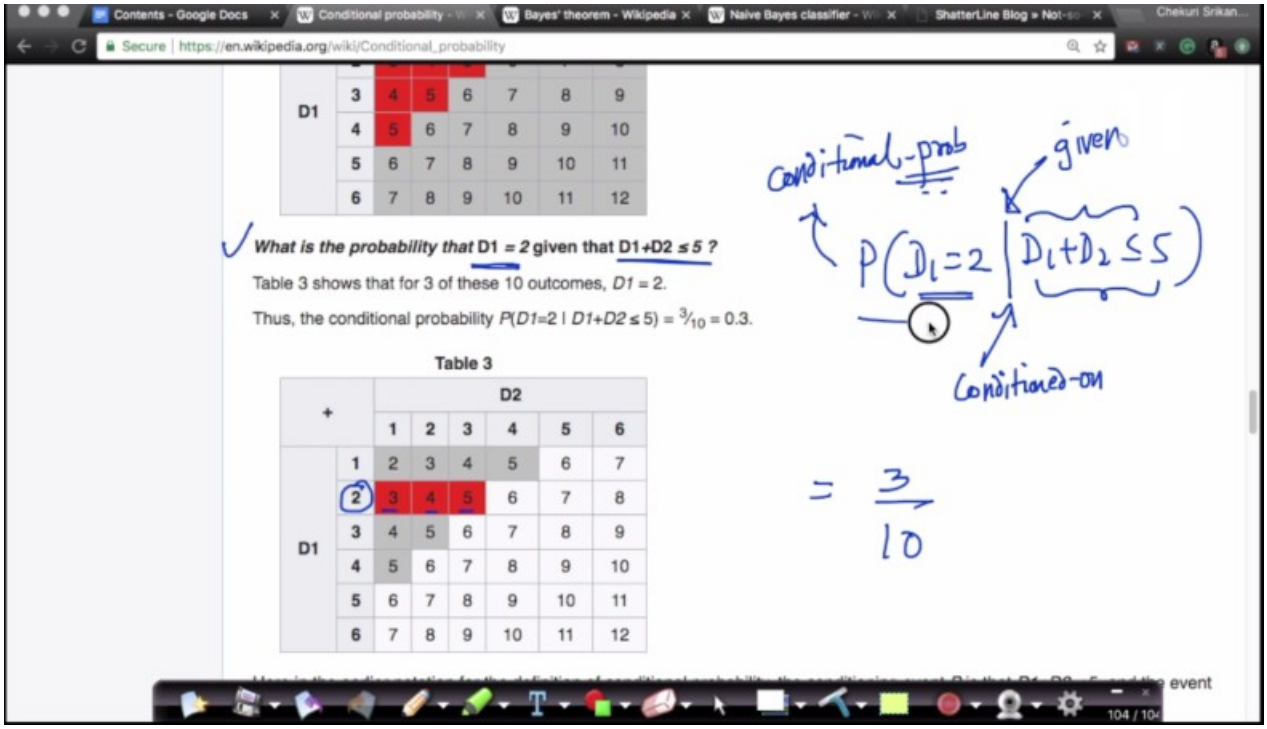



Each cell the sum of outcomes d1 and d2 (dies).
Now, How many possibilities have the sum is less than 5 int he table.

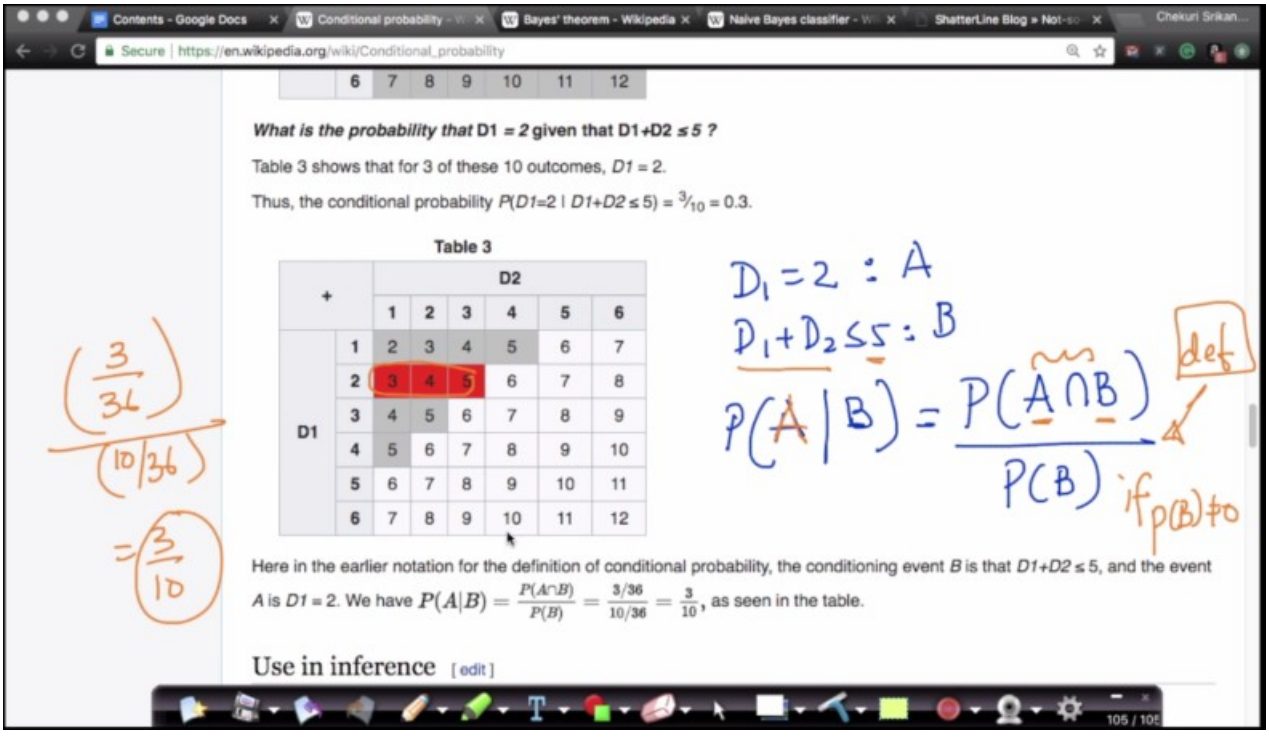


Q. Then what is the probability that d1 = 2 given the condition d1 + d2 <= 5.
A. The answer is nothing but the probability of d1=2 and d1 + d2 is less than 5.

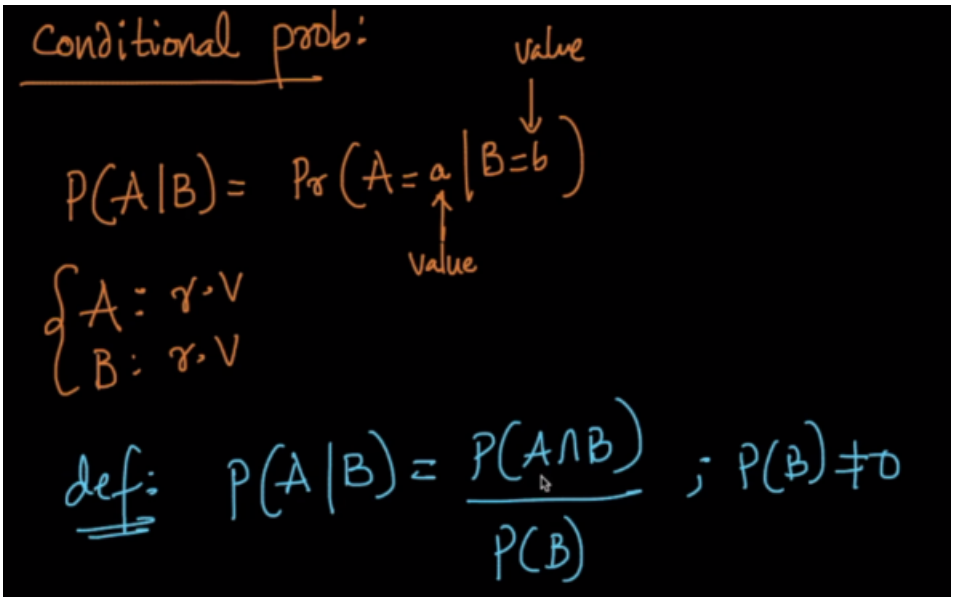
This is called conditional probability as the probability of some thing, when some thing else is already happend.(or given already).




Computing conditional probability:

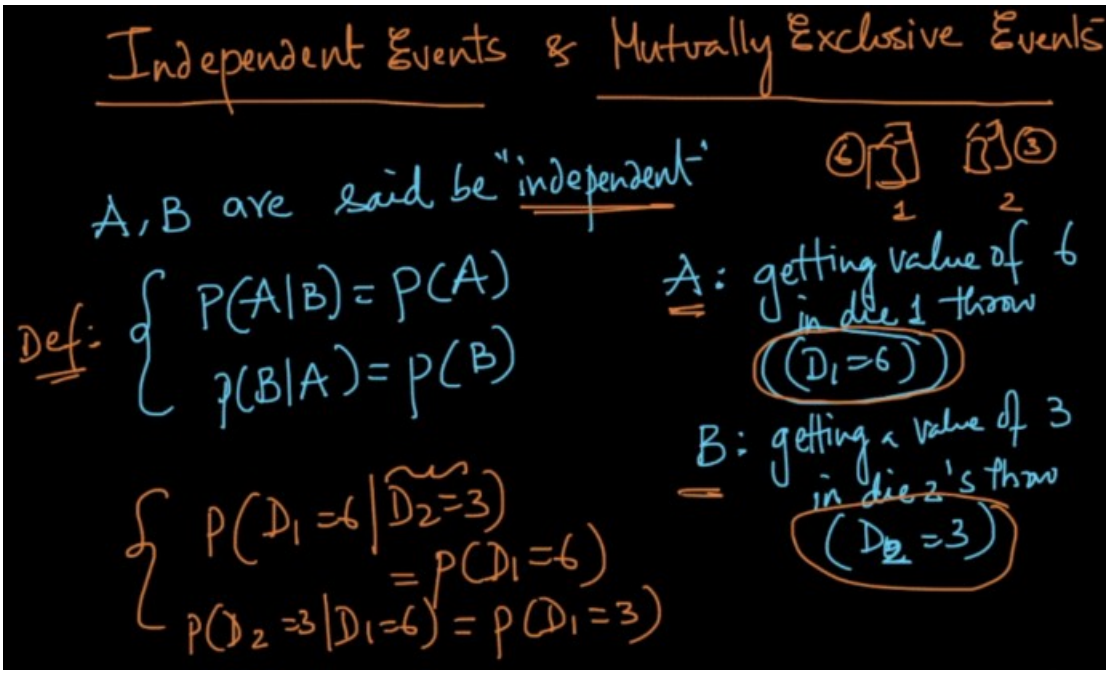


Defining conditional probability:




Independent events and mutually exclusive events:

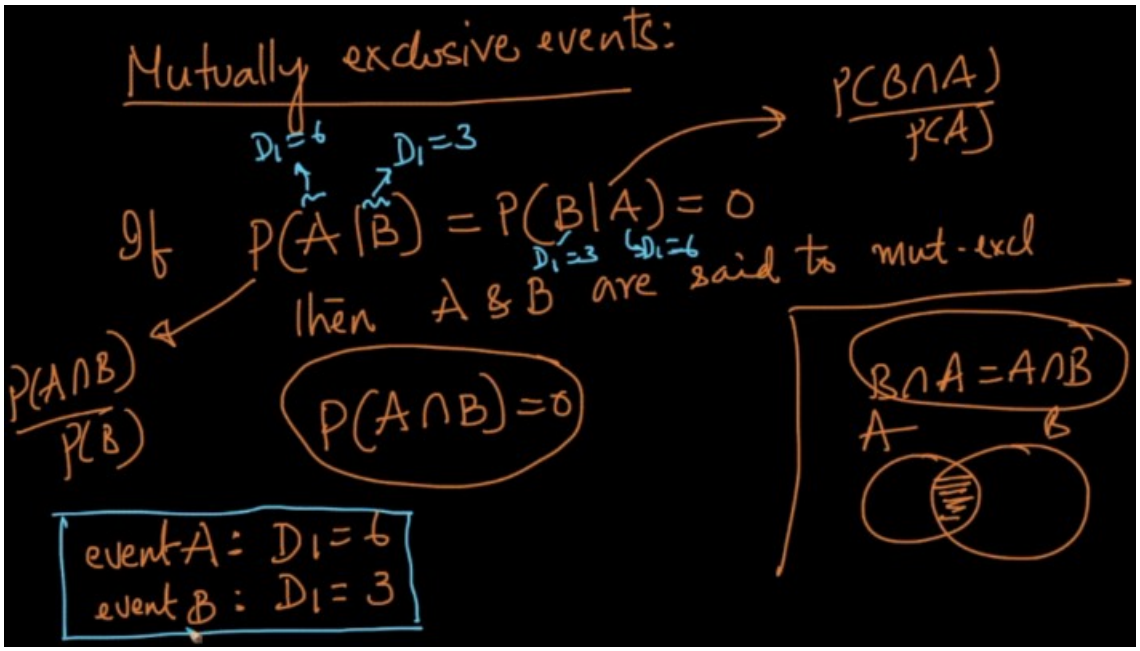
The events are said to be independent, if the occurance of one event does not depend on the occurance of other event.



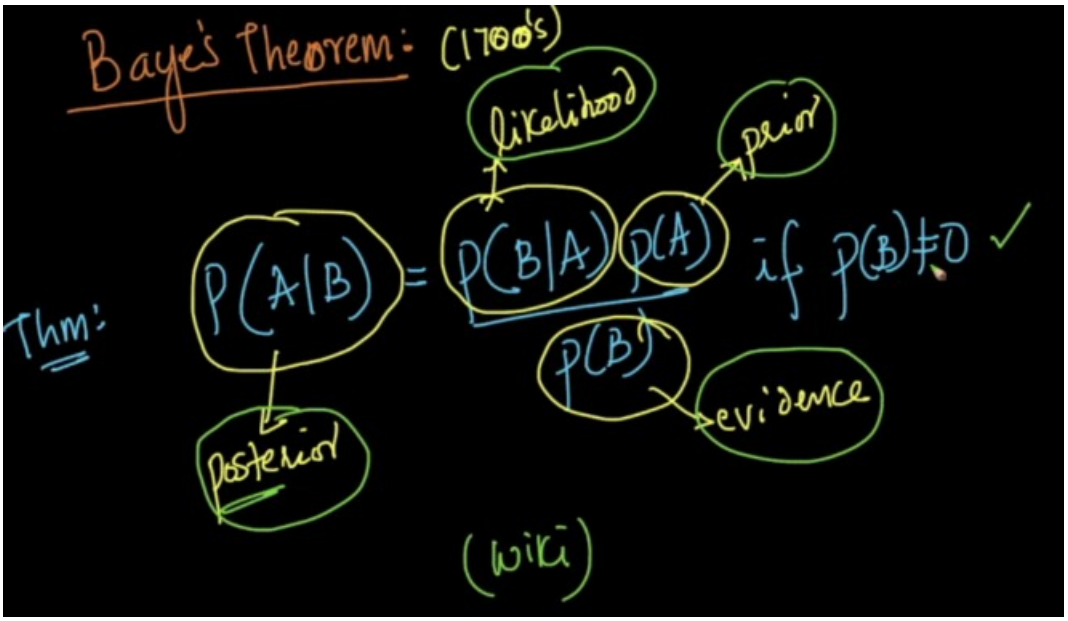

Here is the mathematical derivation of independent events.

Mutually exclusive events:

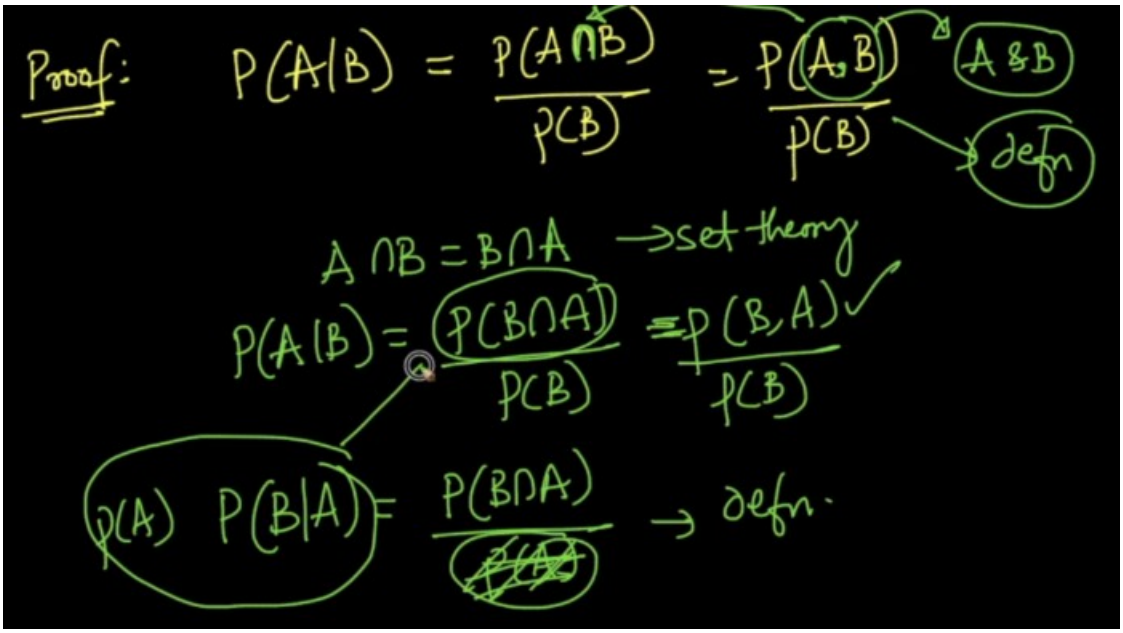
We can understand this concept as the events that are occuring at the same time of one die only. Which is zero, because the probability of occurance of 3 is noting to do with the occurance of the 6 of the same die.



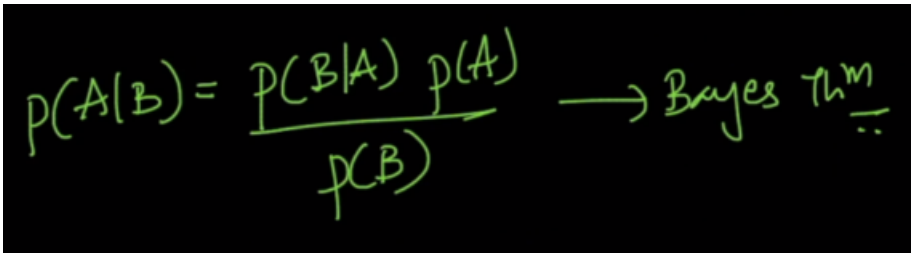

Bayes theorm:
Defining bayes theoem:



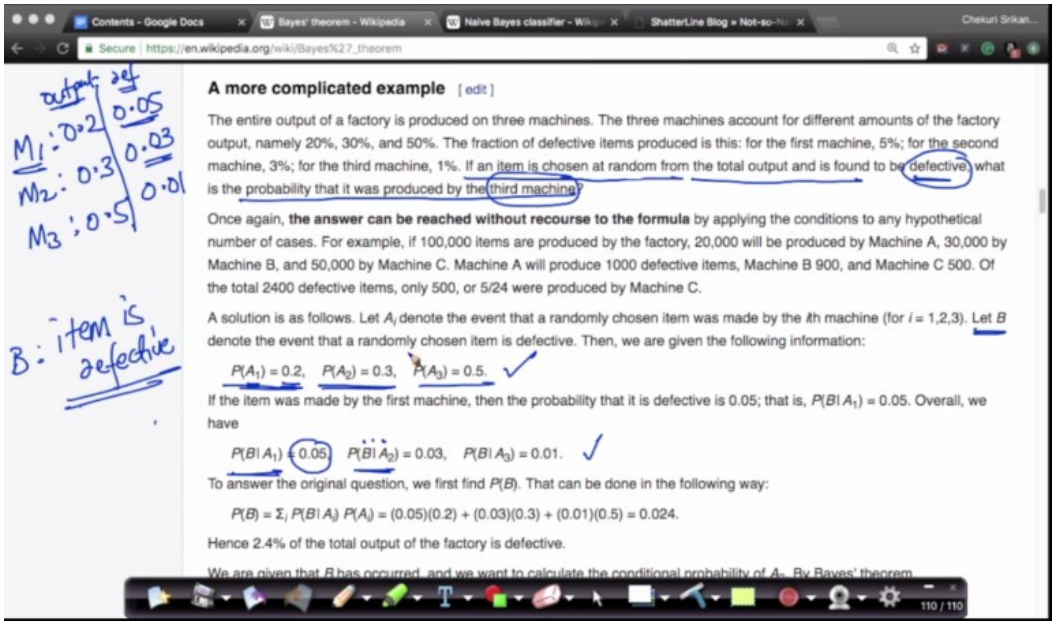
Proof:



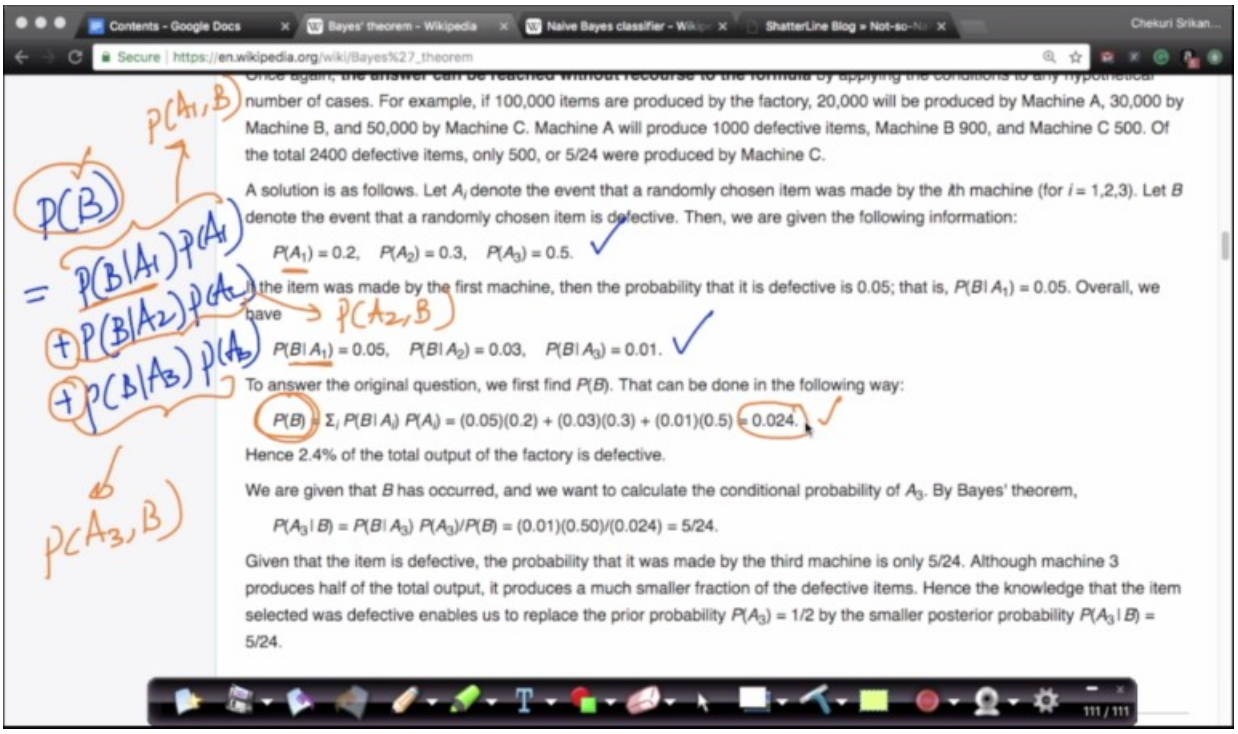

Interchanging the equations of conditional probability, we get the bayes theorm.




Example:



Calculating the probability of defective of the total factory output..



number of cases. For example, if 100,000 items are produced by the factory, 20,000 will be produced by Machine A, 30,000 by Machine B, and 50,000 by Machine C. Machine A will produce 1000 defective items, Machine B 900, and Machine C 500. Of the total 2400 defective items, only 500, or 5/24 were produced by Machine C.

A solution is as follows. Let $A_i$ denote the event that a randomly chosen item was made by the $i$th machine (for $i = 1,2,3$). Let $B$ 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, $P(B|A_1) = 0.05$. Overall, we have

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

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

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

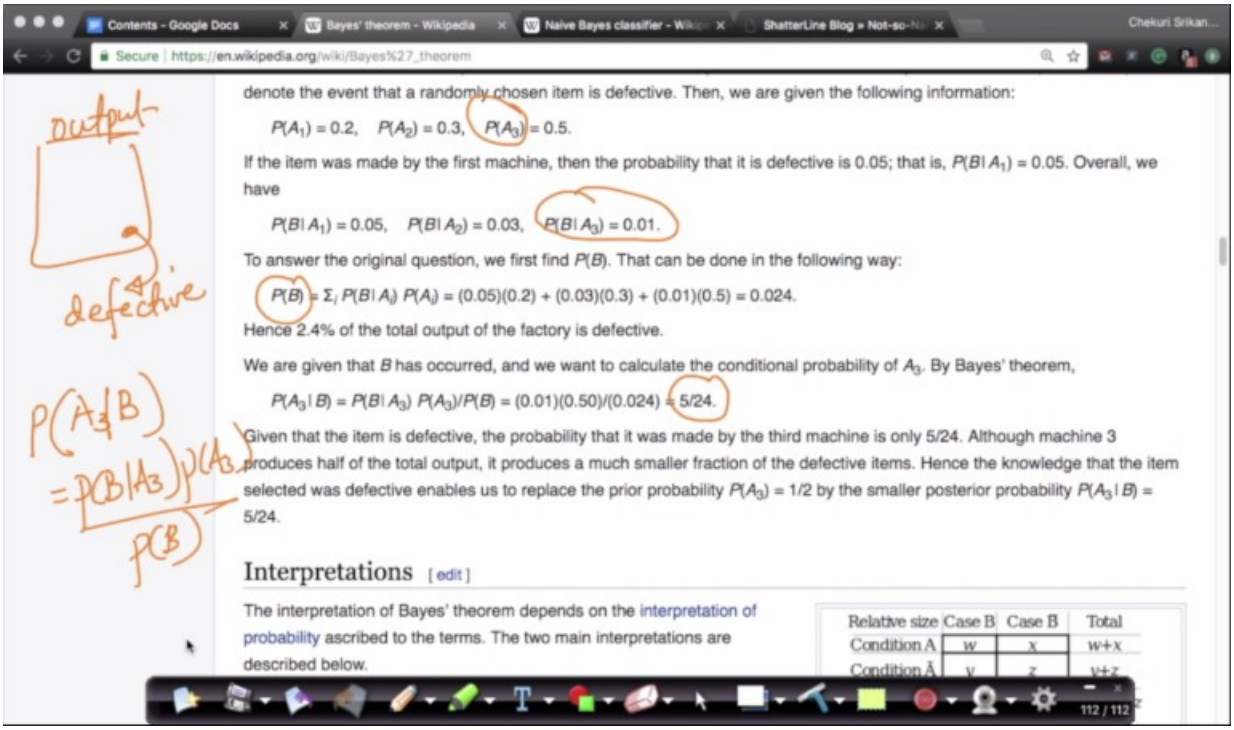
Hence 2.4% of the total output of the factory is defective.

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

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

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 $P(A_3|B) = 5/24$.

Next, the probability of the randomly chosen item belongs to the machine A3.



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, $P(B|A_1) = 0.05$. Overall, we have

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

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

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

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,

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

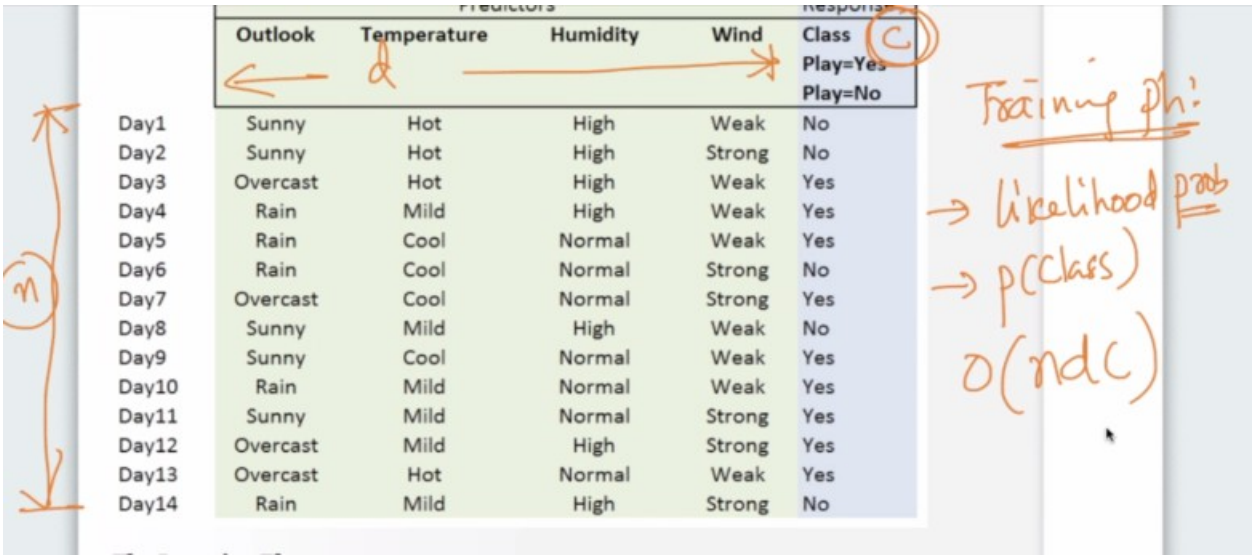
Given that the item is defective, the probability that it was made by the third machine is only 5/24. Although machine 3 produces half of the total output, it produces a much smaller fraction of the defective items. Hence the knowledge that the item selected was defective enables us to replace the prior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

## Interpretations [edit]

The interpretation of Bayes' theorem depends on the interpretation of probability ascribed to the terms. The two main interpretations are described below.

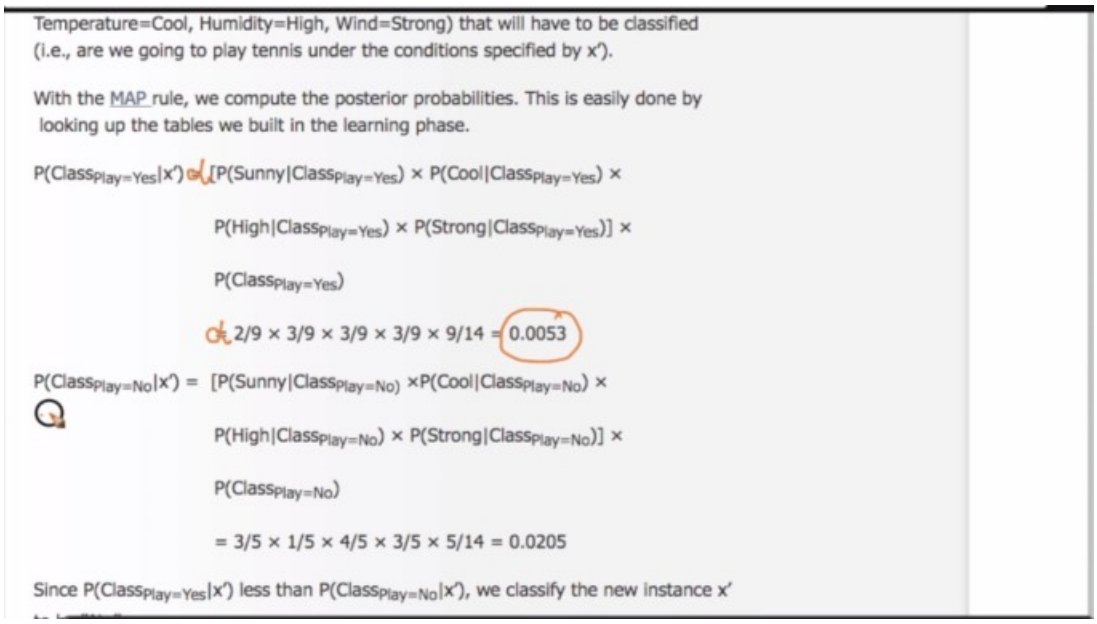
| Relative size | Case B | Case B | Total |
|---|---|---|---|
| Condition A | $w$ | $x$ | $w+x$ |
| Condition Ā | $y$ | $z$ | $y+z$ |

Data:



| | Outlook | Temperature | Humidity | Wind | Class Play=Yes / Play=No |
|---|---|---|---|---|---|
| Day1 | Sunny | Hot | High | Weak | No |
| Day2 | Sunny | Hot | High | Strong | No |
| Day3 | Overcast | Hot | High | Weak | Yes |
| Day4 | Rain | Mild | High | Weak | Yes |
| Day5 | Rain | Cool | Normal | Weak | Yes |
| Day6 | Rain | Cool | Normal | Strong | No |
| Day7 | Overcast | Cool | Normal | Strong | Yes |
| Day8 | Sunny | Mild | High | Weak | No |
| Day9 | Sunny | Cool | Normal | Weak | Yes |
| Day10 | Rain | Mild | Normal | Weak | Yes |
| Day11 | Sunny | Mild | Normal | Strong | Yes |
| Day12 | Overcast | Mild | High | Strong | Yes |
| Day13 | Overcast | Hot | Normal | Weak | Yes |
| Day14 | Rain | Mild | High | Strong | No |

We can take the propotionality from the naive bayes and derive the LHS of the naive bayes propotional to the numerator of Naive bayes.
Example:



Temperature=Cool, Humidity=High, Wind=Strong) that will have to be classified (i.e., are we going to play tennis under the conditions specified by x').

With the MAP rule, we compute the posterior probabilities. This is easily done by looking up the tables we built in the learning phase.

$P(Class_{Play=Yes}|x') \propto [P(Sunny|Class_{Play=Yes}) \times P(Cool|Class_{Play=Yes}) \times$

$P(High|Class_{Play=Yes}) \times P(Strong|Class_{Play=Yes})] \times$

$P(Class_{Play=Yes})$

$\propto 2/9 \times 3/9 \times 3/9 \times 3/9 \times 9/14 = 0.0053$

$P(Class_{Play=No}|x') = [P(Sunny|Class_{Play=No}) \times P(Cool|Class_{Play=No}) \times$

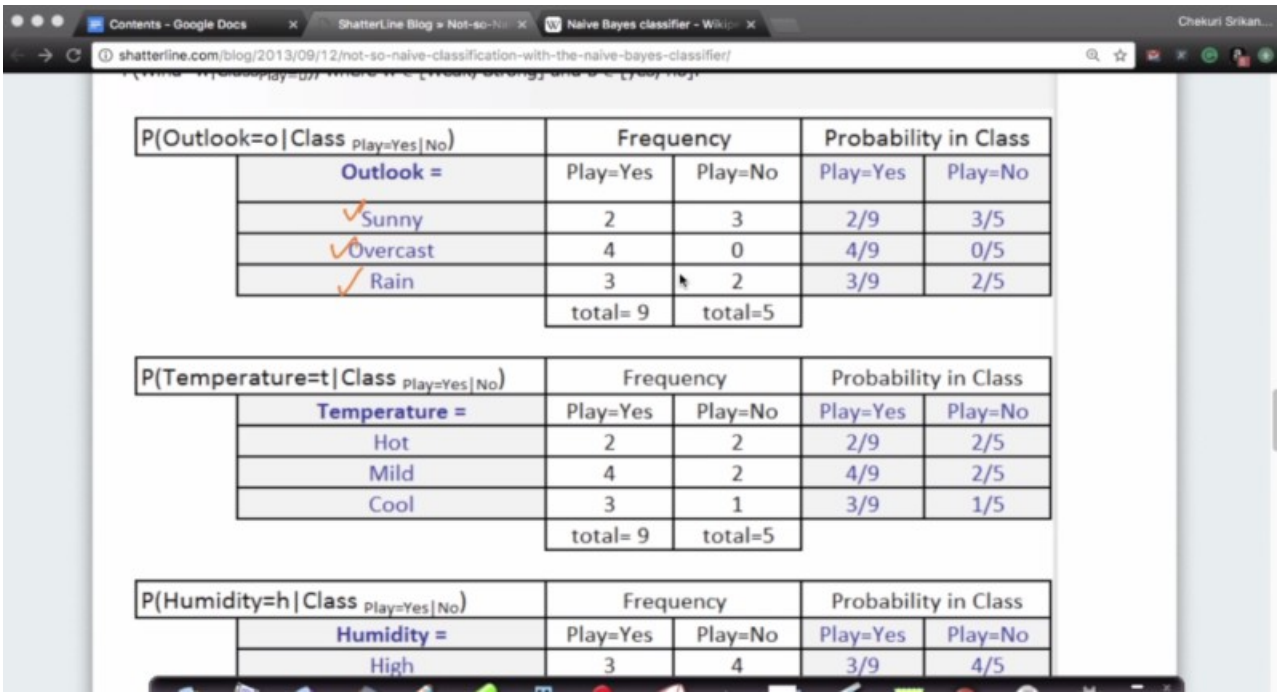
$P(High|Class_{Play=No}) \times P(Strong|Class_{Play=No})] \times$

$P(Class_{Play=No})$

$= 3/5 \times 1/5 \times 4/5 \times 3/5 \times 5/14 = 0.0205$

Since $P(Class_{Play=Yes}|x')$ less than $P(Class_{Play=No}|x')$, we classify the new instance x'

Contengency tables:



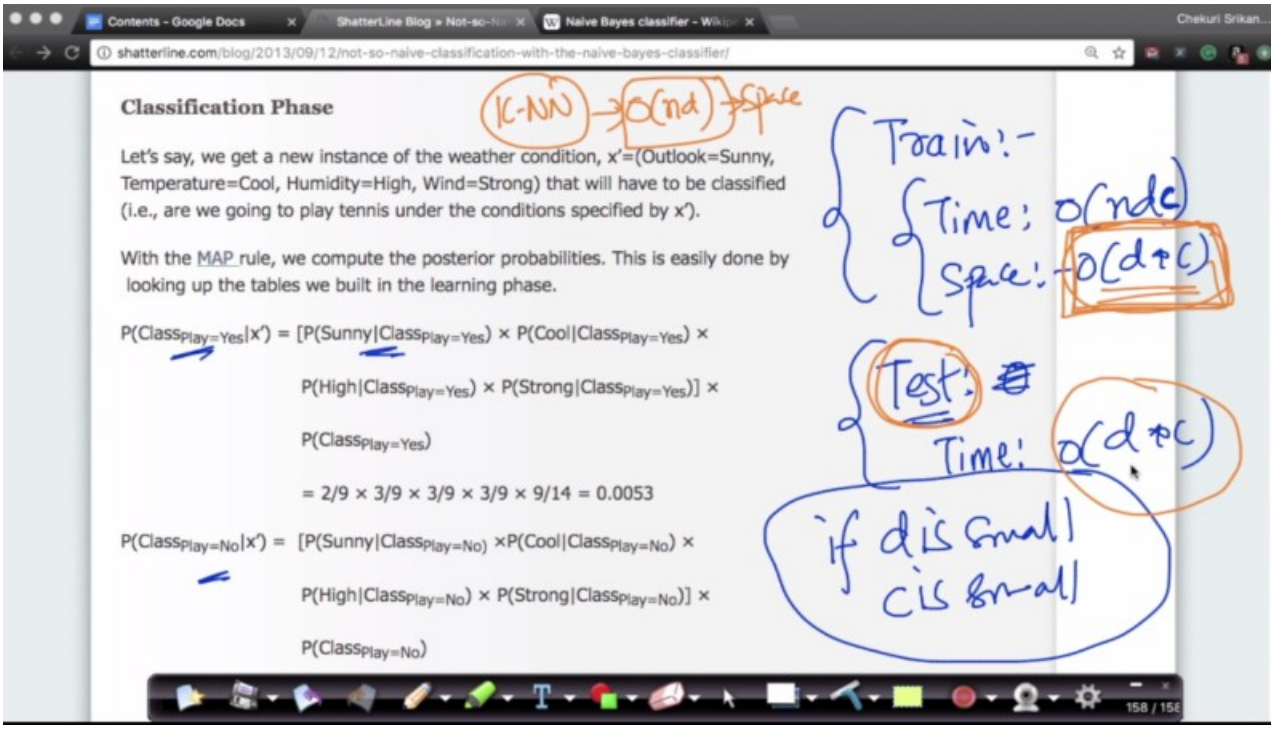
| P(Outlook=o\|Class Play=Yes\|No) | Frequency | | Probability in Class | |
|---|---|---|---|---|
| Outlook = | Play=Yes | Play=No | Play=Yes | Play=No |
| Sunny | 2 | 3 | 2/9 | 3/5 |
| Overcast | 4 | 0 | 4/9 | 0/5 |
| Rain | 3 | 2 | 3/9 | 2/5 |
| | total= 9 | total=5 | | |

| P(Temperature=t\|Class Play=Yes\|No) | Frequency | | Probability in Class | |
|---|---|---|---|---|
| Temperature = | Play=Yes | Play=No | Play=Yes | Play=No |
| Hot | 2 | 2 | 2/9 | 2/5 |
| Mild | 4 | 2 | 4/9 | 2/5 |
| Cool | 3 | 1 | 3/9 | 1/5 |
| | total= 9 | total=5 | | |

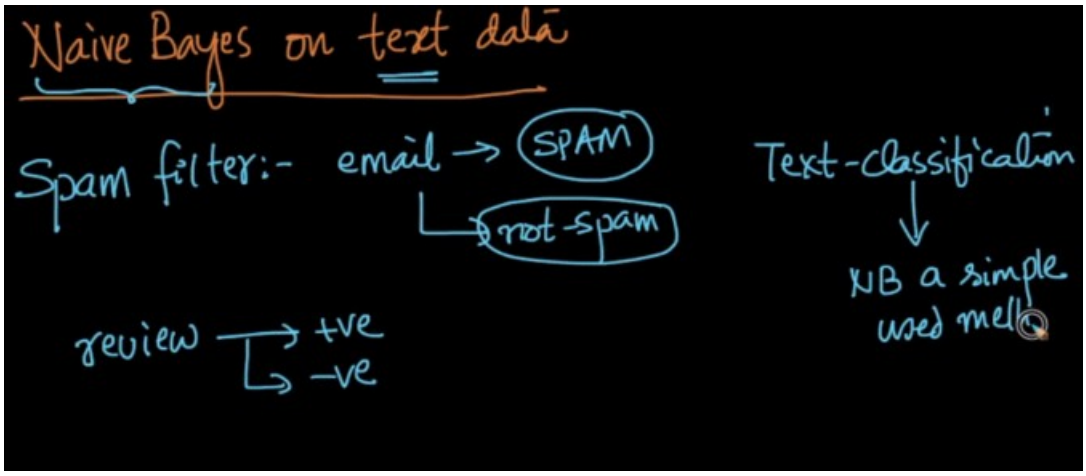
| P(Humidity=h\|Class Play=Yes\|No) | Frequency | | Probability in Class | |
|---|---|---|---|---|
| Humidity = | Play=Yes | Play=No | Play=Yes | Play=No |
| High | 3 | 4 | 3/9 | 4/5 |

The time complexity during the traning time is O(nd). Then we can store the probability values in the dictionary and retrive during the test time.

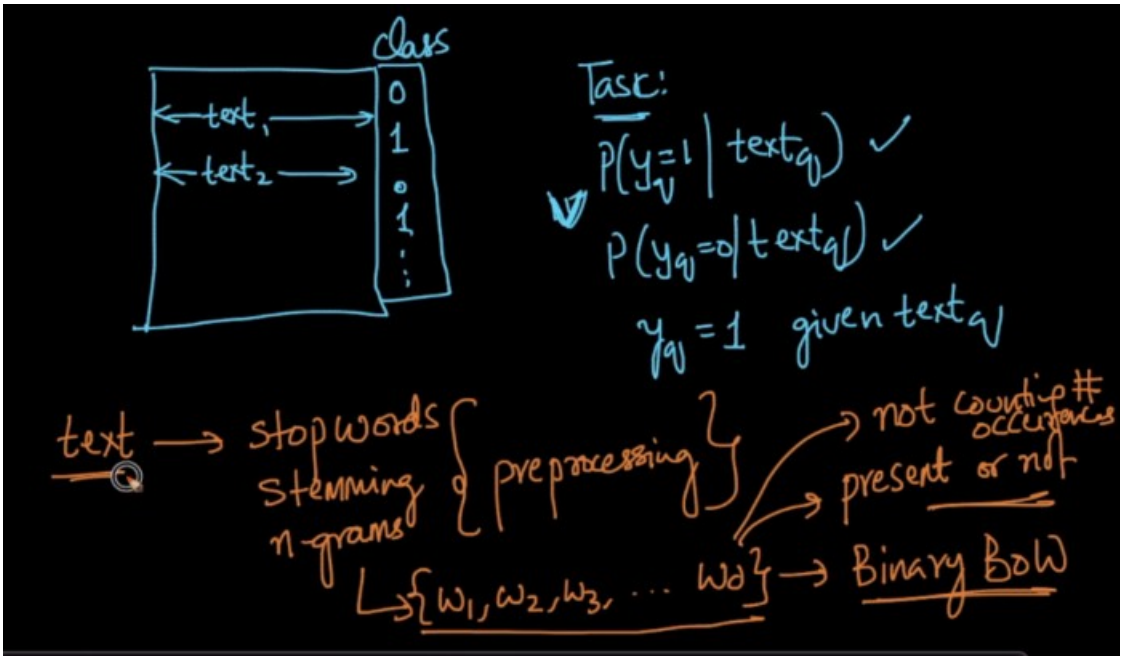
In case of categorical features, implementing naive bayes in straight forward aka **counting** the unique values of the features.
Train and time complexity:



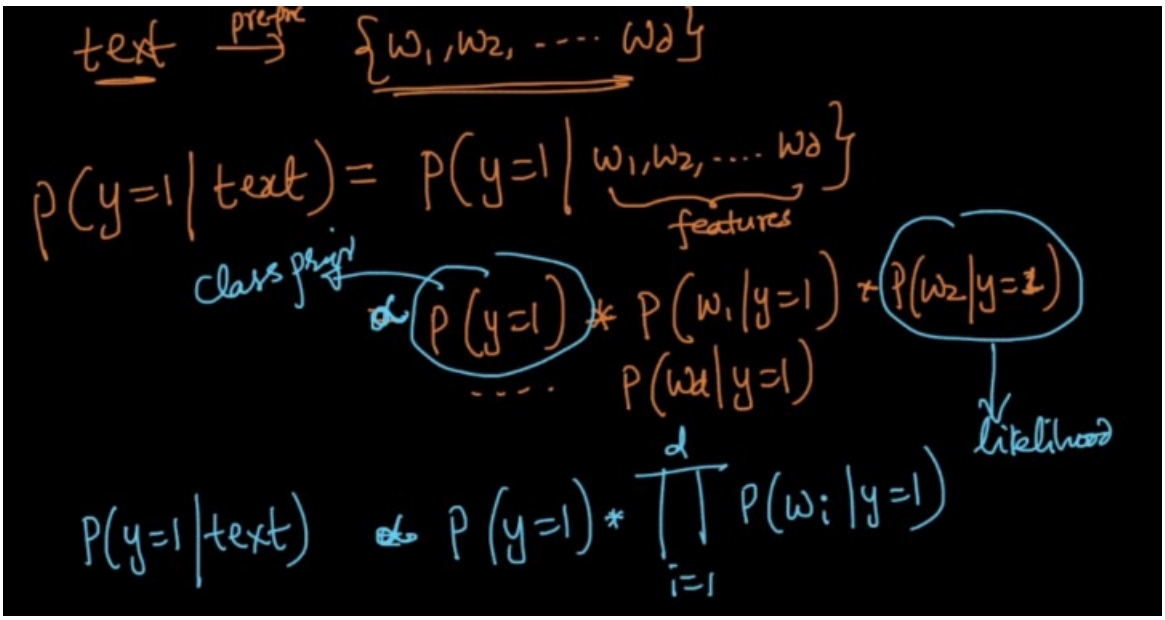

Naive Bayes on text data:
NB is often used for text classification, spam filtering.



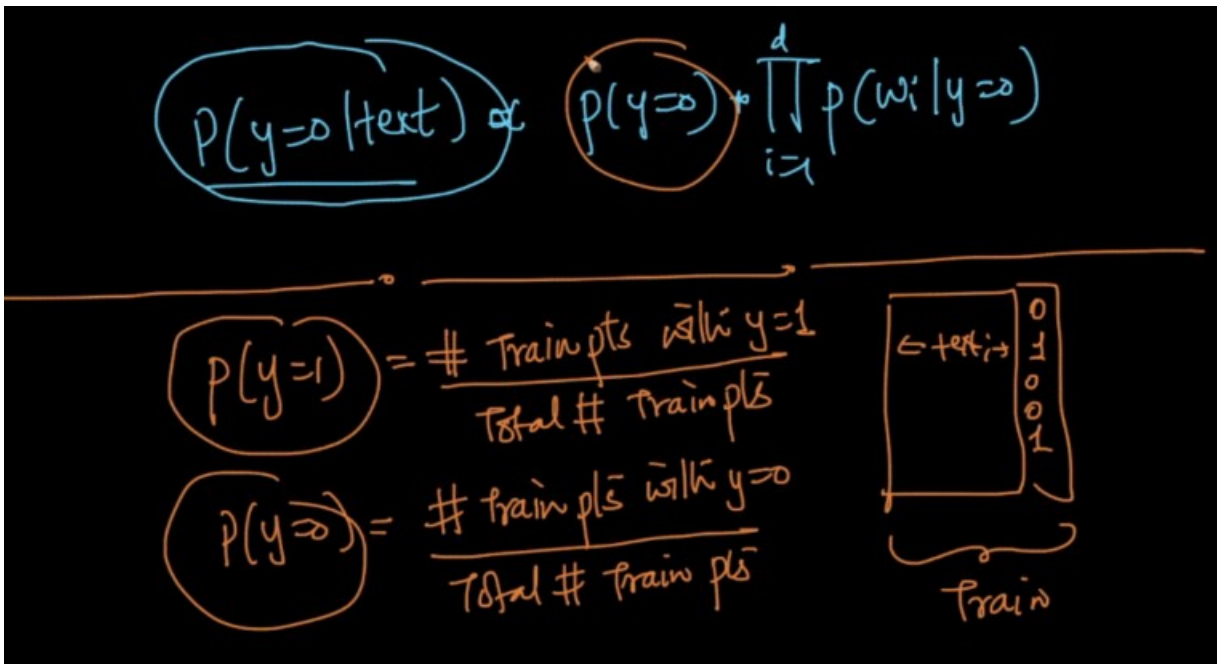

The binary bag of words of text data is obtained by most common text preprocessing techniques.




The probability of text is spam can be written as follows:

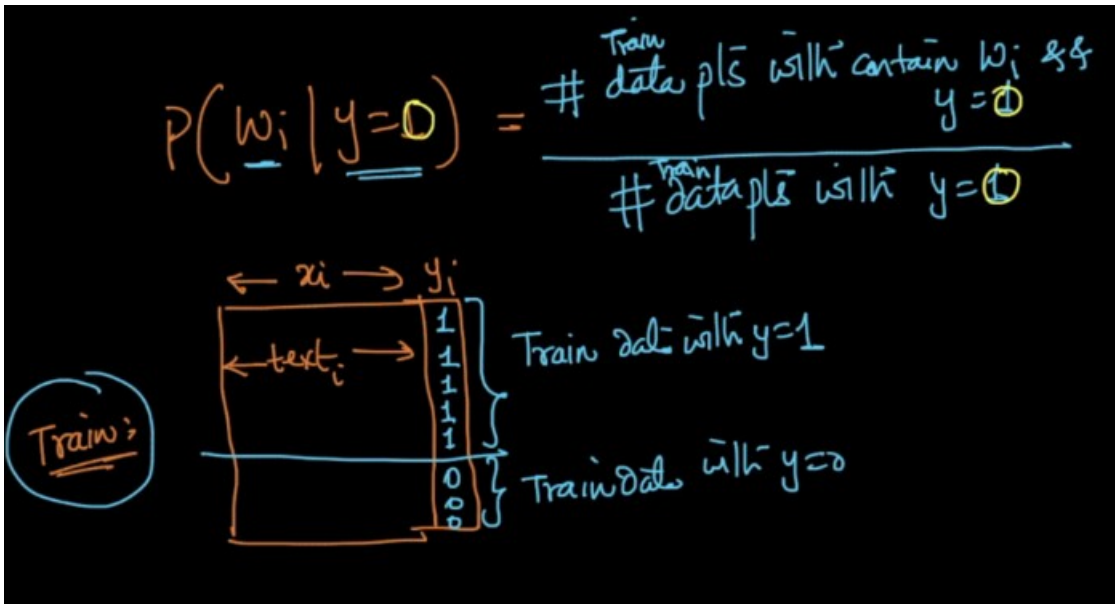



The words are features in this case.

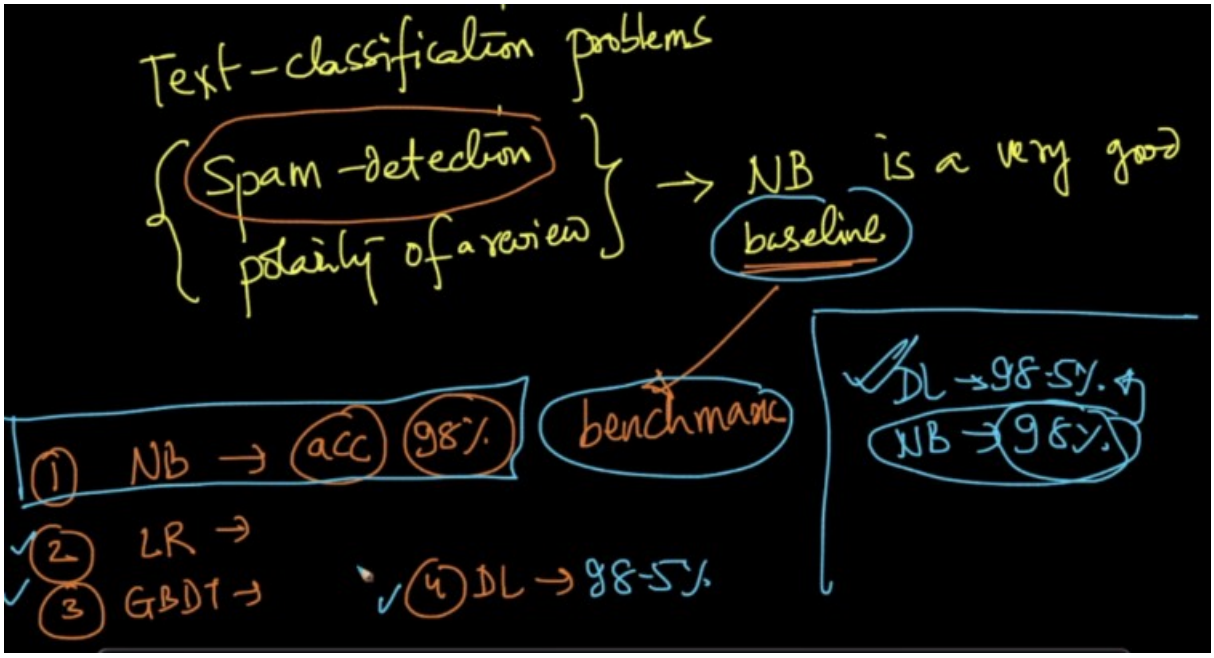



The below part of the snap is the calculation of **prior** probabilities.

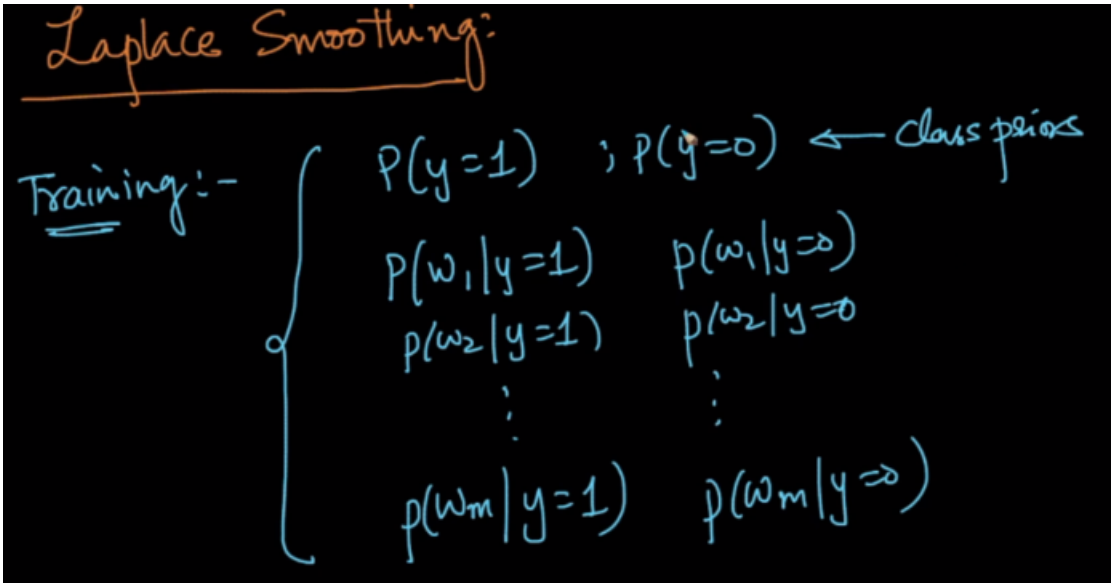
Calculation of liklihood:



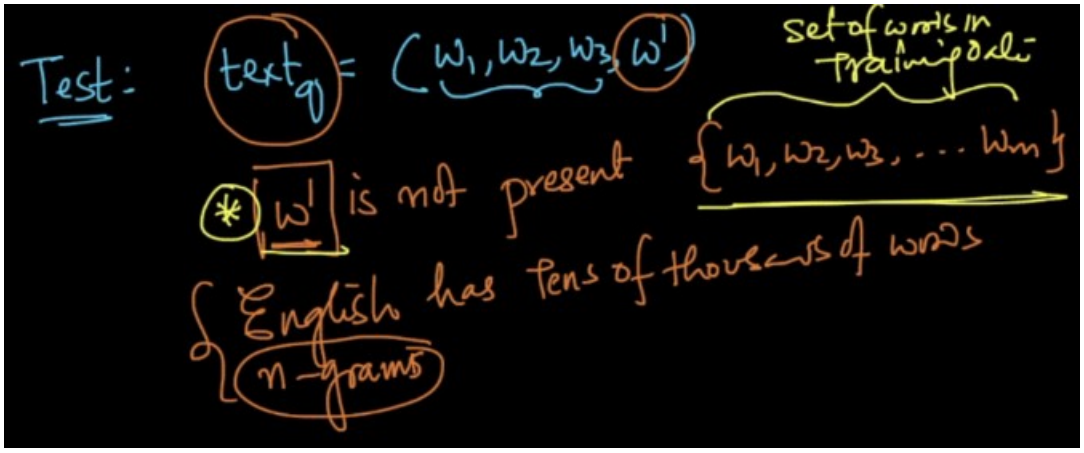

**Naive bayes is often used as a baseline model for improving the accuracy.**



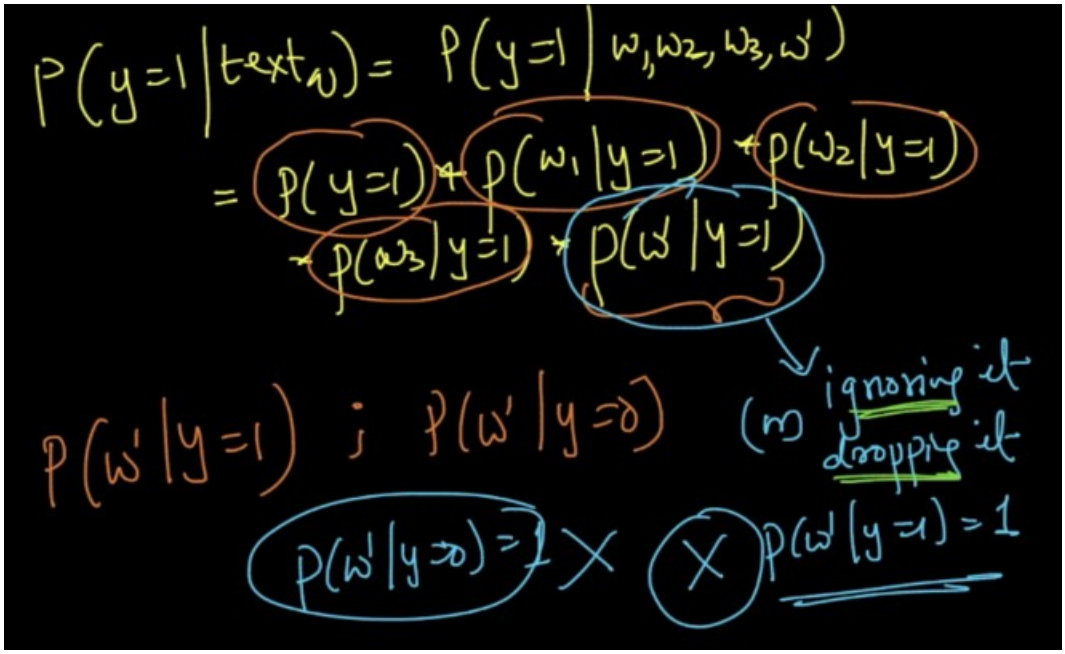

Laplace and additive smoothing:
Summary of training the models:




During the text time if the words does not occur in the traning data.

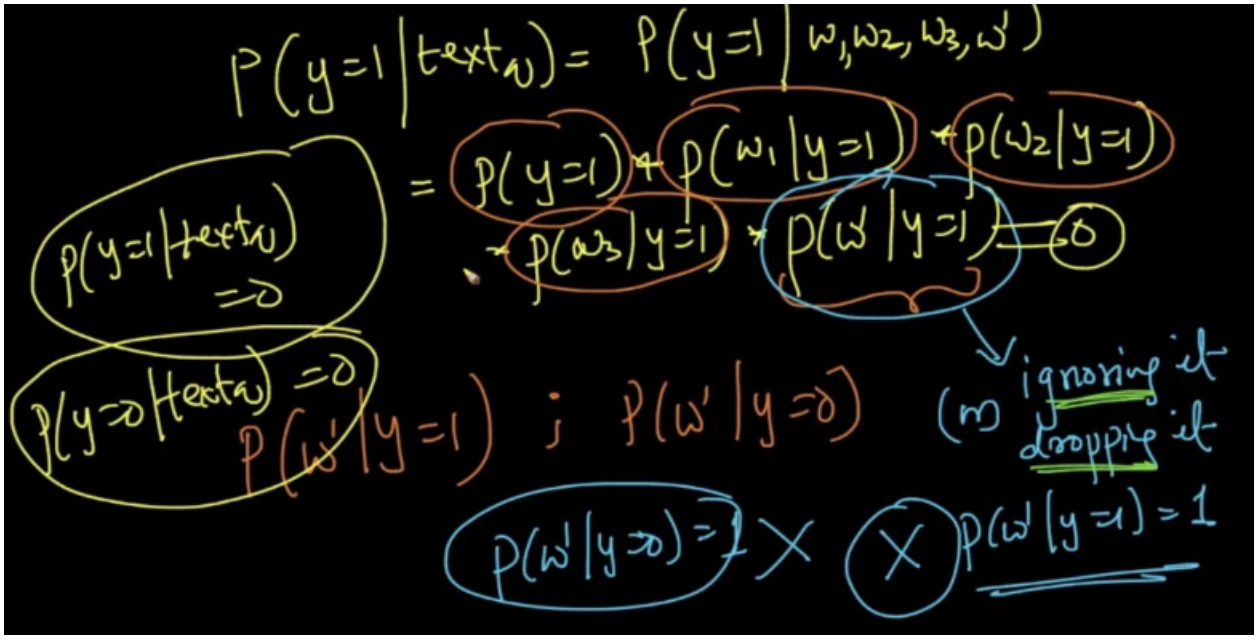



Questioning the model, with test data point.

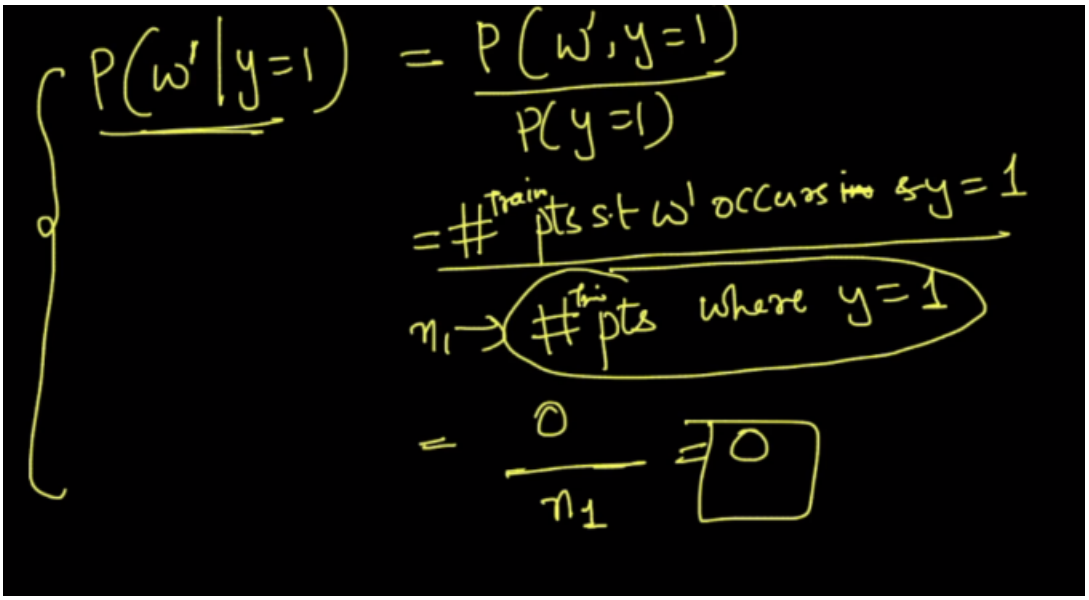



Ignoring the word that is not there in the traning data is nothing but making the word probability equal to 1. This is logically incorrect.

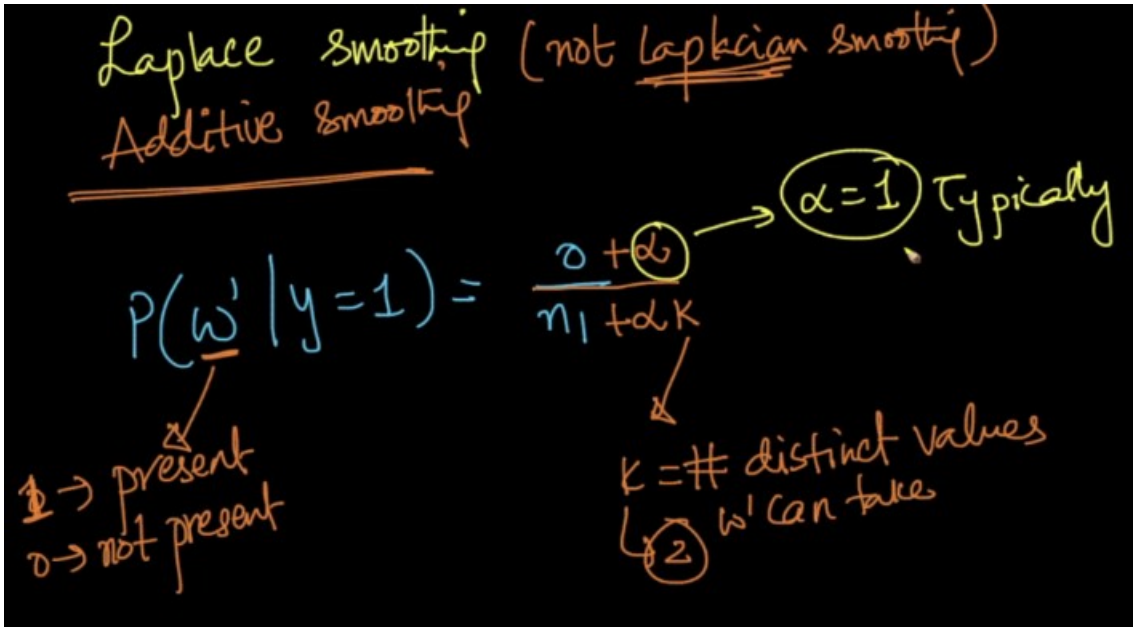
Making the probability of occurance of the word to zero will make the whole probability reduce to zero.




Here is the intution of the missing word problem, the probability reduces to zero.



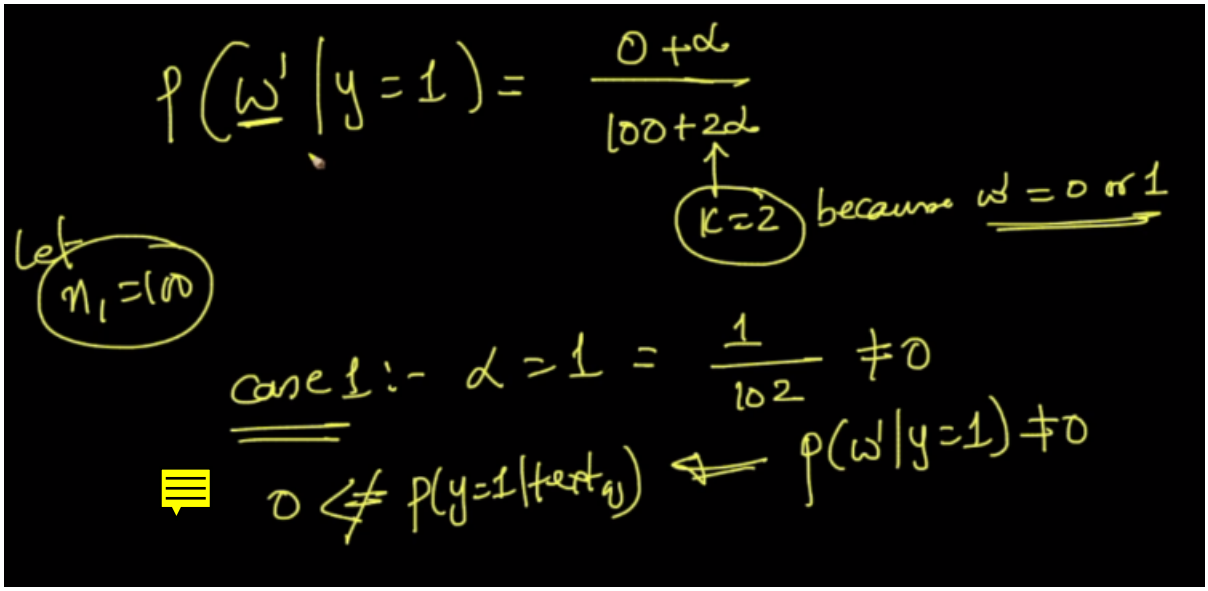

To overcome this problem, we use something called laplace smoothing / additive smoothing.




Here K is the distinct values the feature can take, like when k=2 it can take the word in spam or not spam.

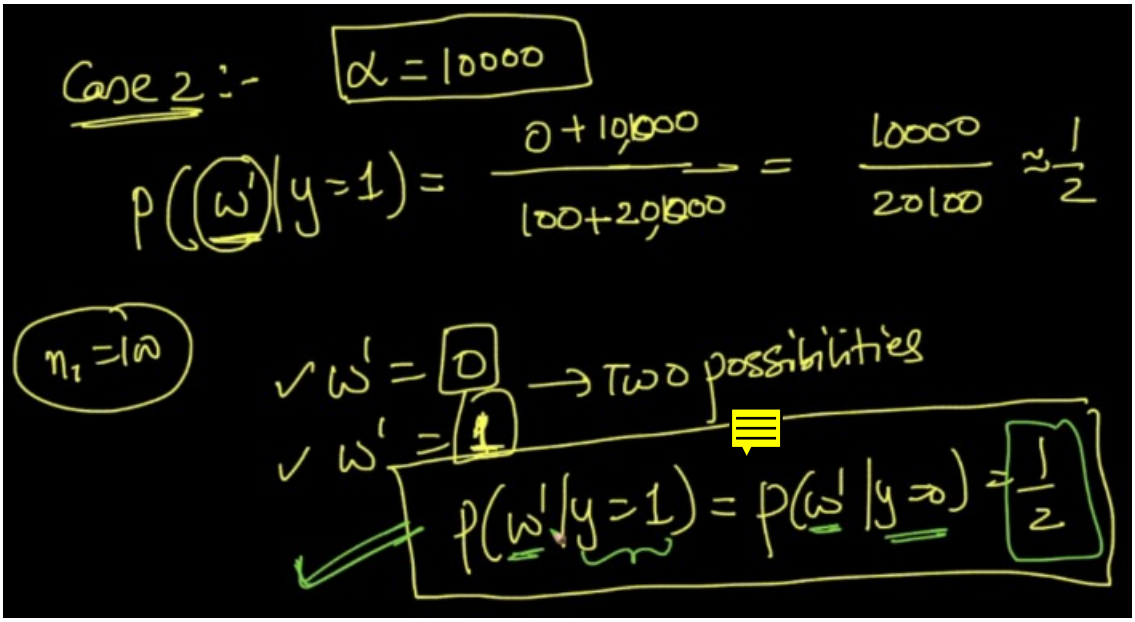
Typically, alpha is taken as 1.

Example:

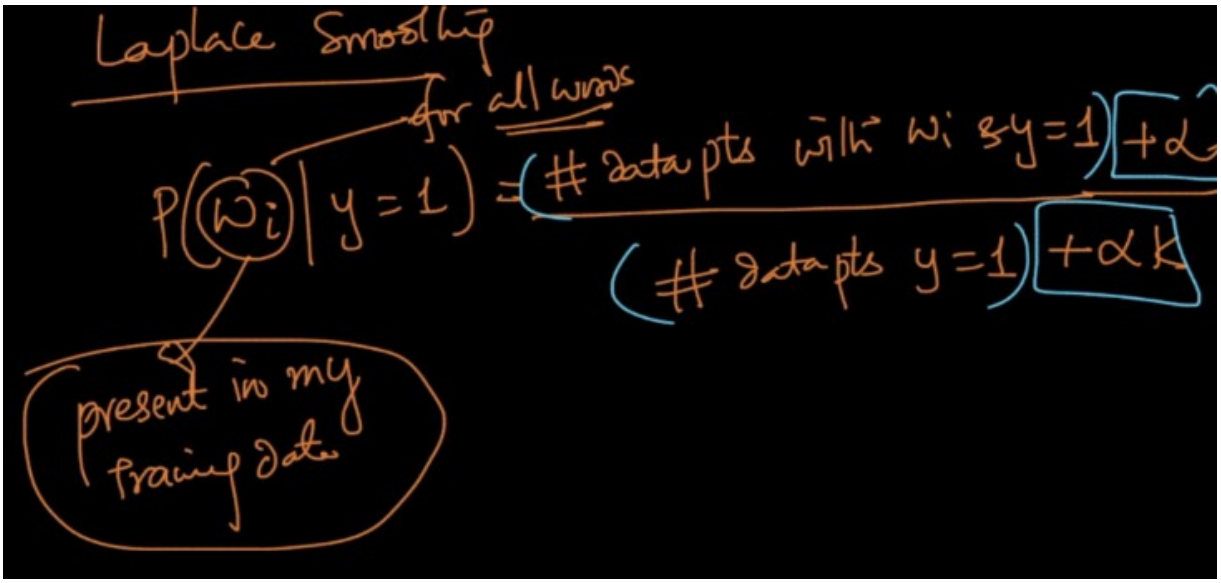


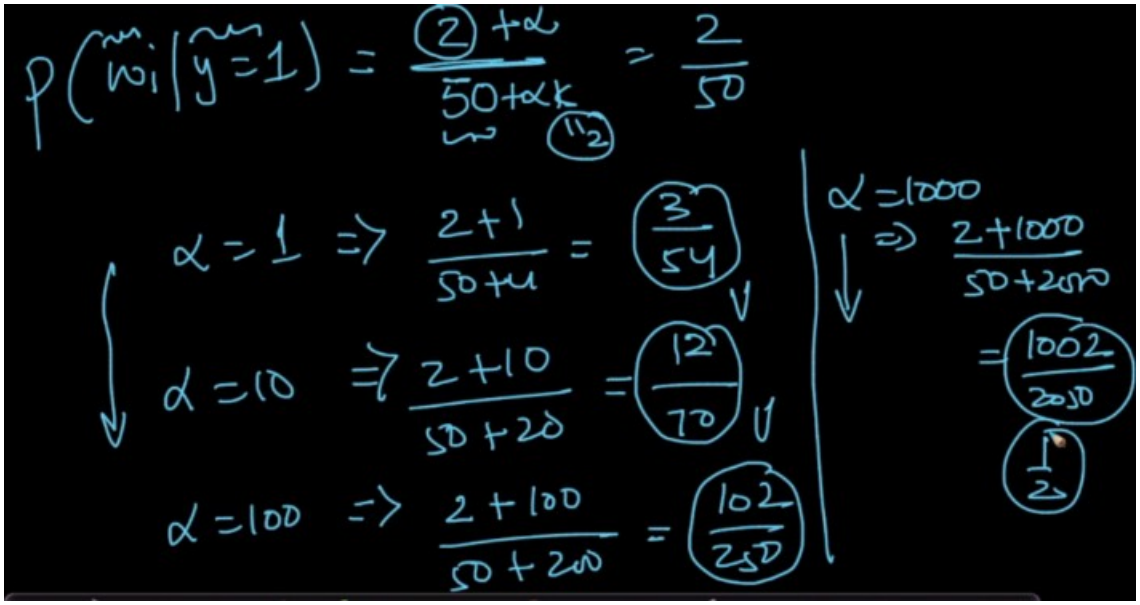

Why dont we just replace with some small value instead of laplace smooting ?

By placing laplace smoothing, we can interpret that the occurance of words that does not occur can be given equi probable, what ever the value of alpha is.




If we apply laplace smoothing every word gets applied by the laplace term. (or) additive – smoothing.





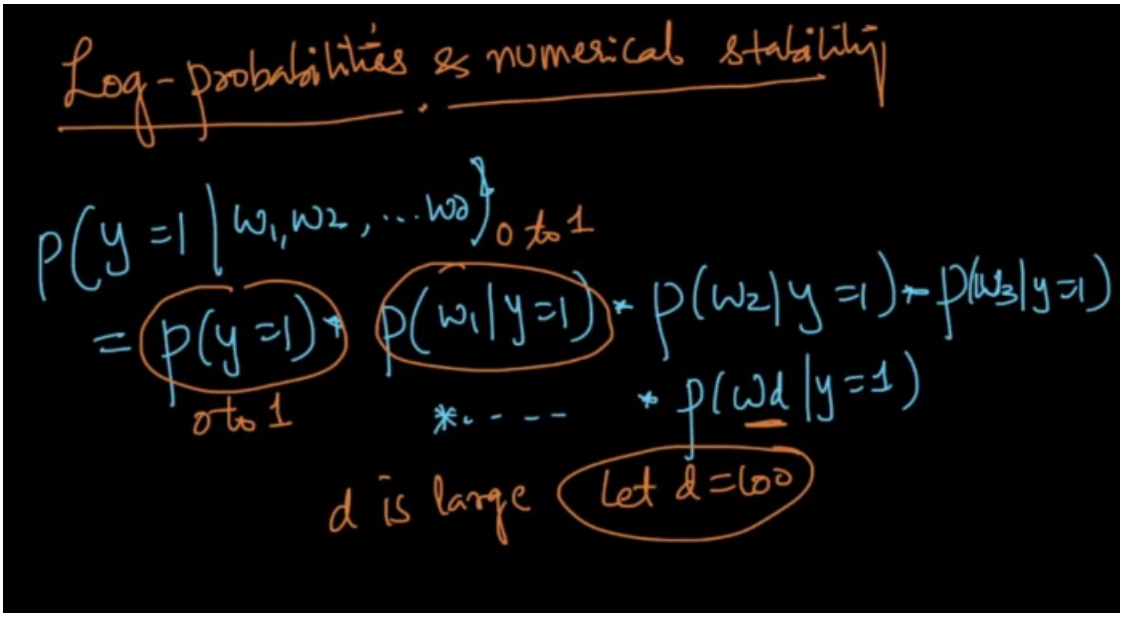

If the alpha increases, the likelihood probabilities are moved to uniform distribution.

If the ratio value is very less, we can apply laplace smoothing to make the probabilities to be equal to half, which is very promising.

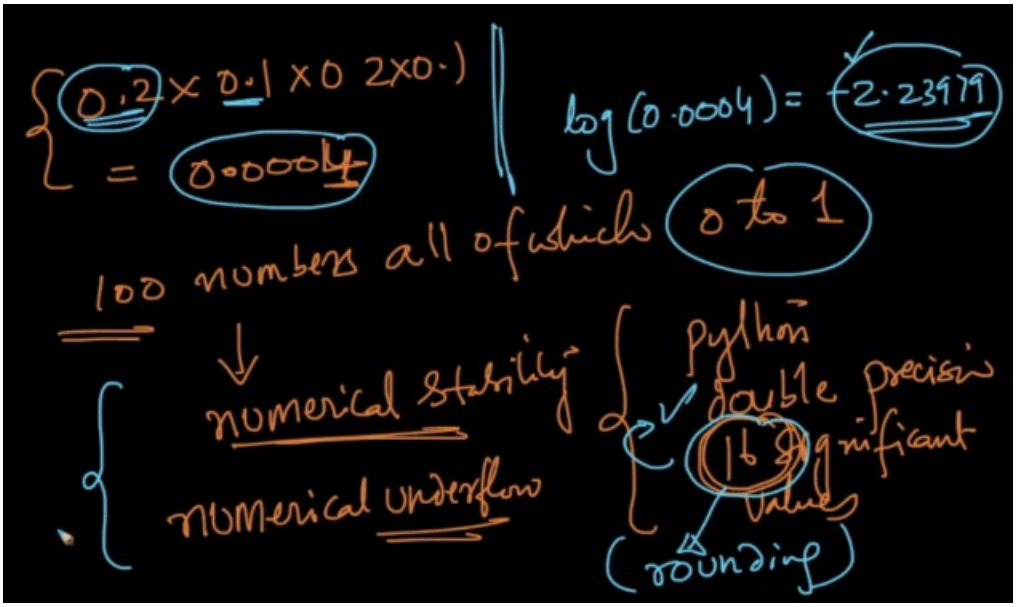
This is called Smooting because, we are eventually moving the values of probabilities to uniform distribution by applying laplace smoothing.

Generally, we apply alpha equal to 1.
Log – probabilities and numerical stability:

For value of d equal to 100. calculation of posterior probabilities.

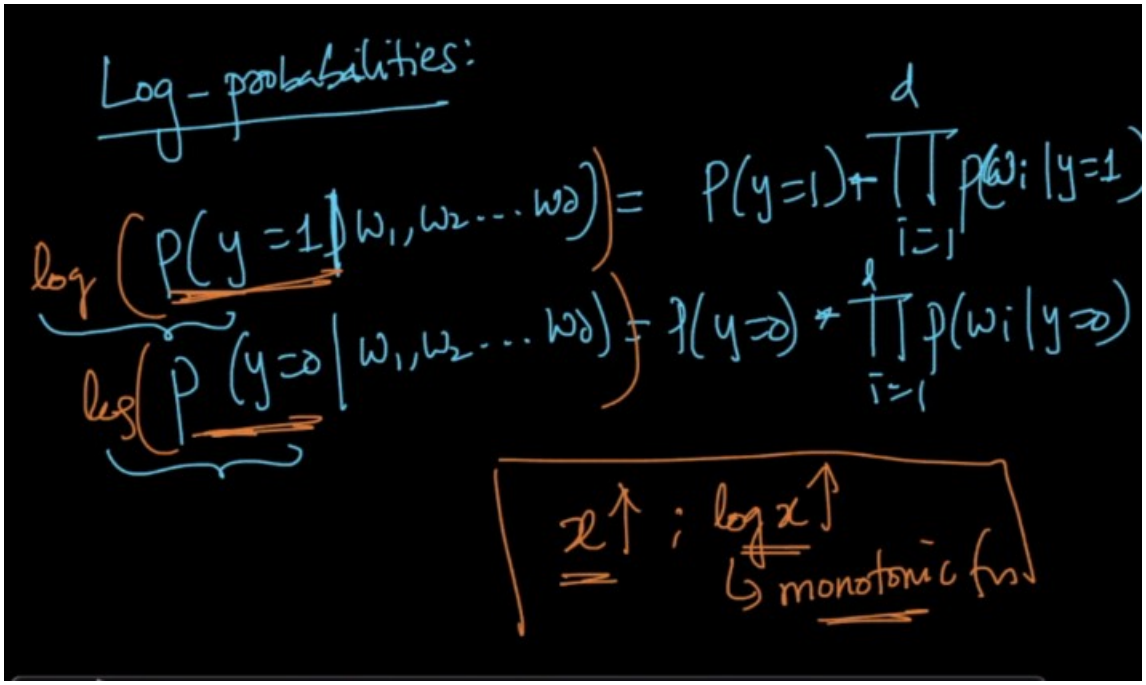



When the values of the probabilities are so small then there is the problem of numerical stability and numberical underflow issues.
Most of the languages will tend to round the values, which will lead to rounding errors.

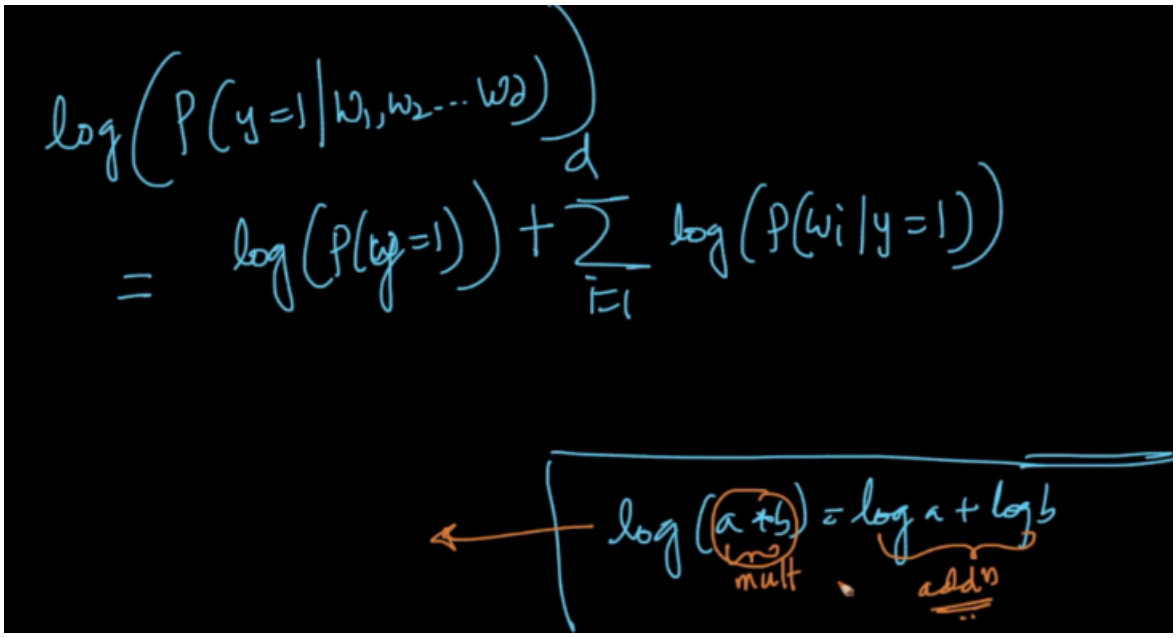



Instead of using these small values, we will use log of each probabilities. It makes more efficent.

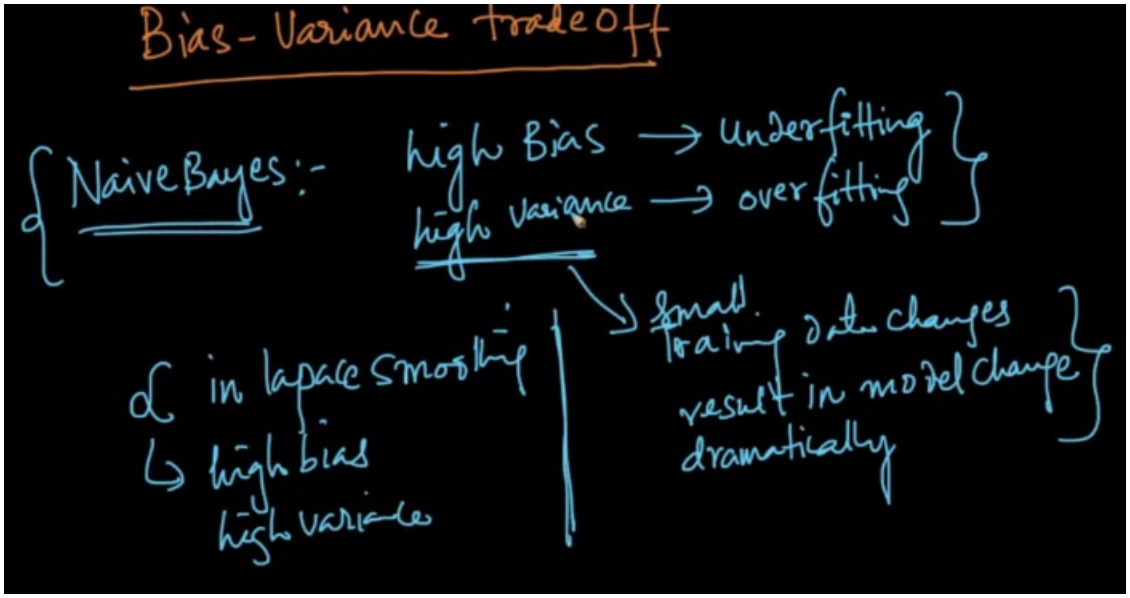
We use log – probabilities:



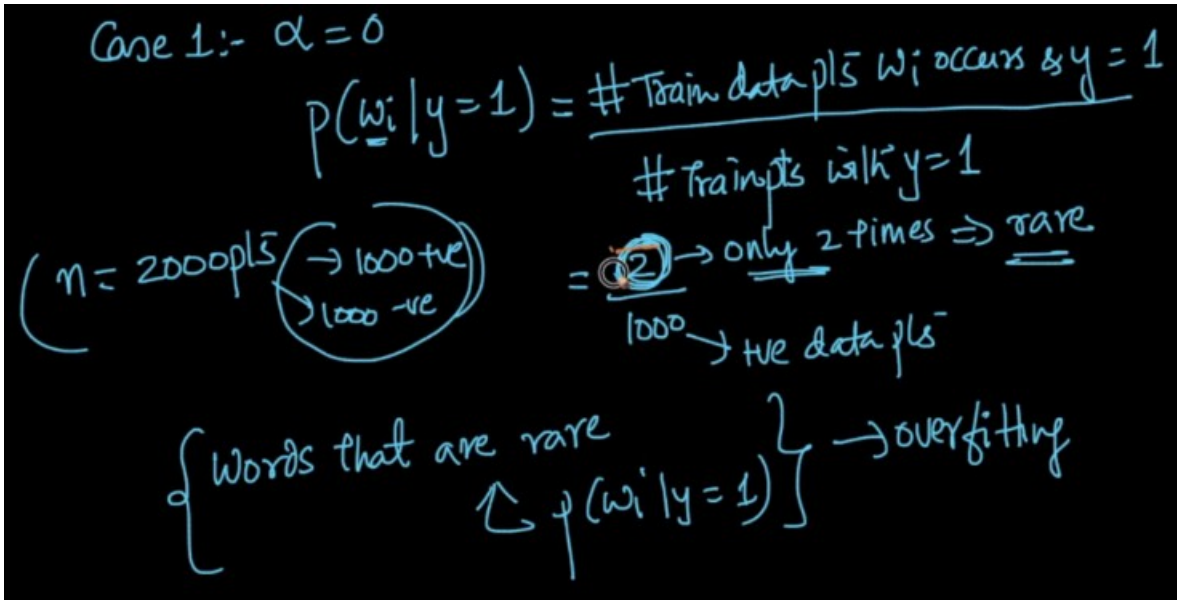

The product becomes summation for log probabilities (basic proberty of the log).



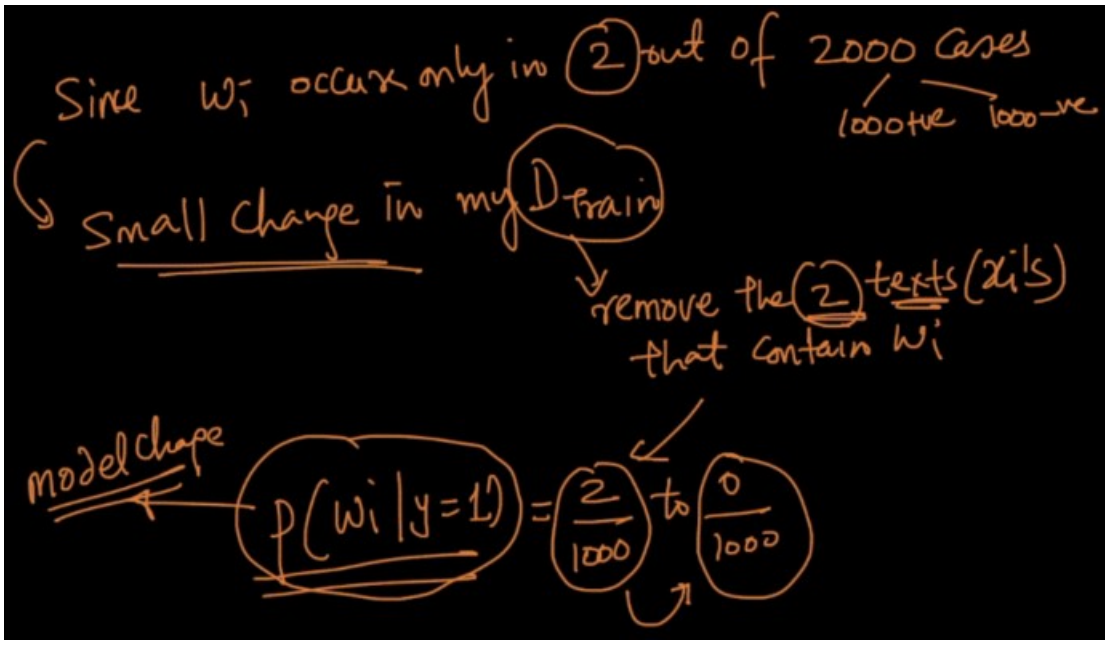
Bias – variance tradeoff:
In naive bayes the value of alpha judges the bias – variance tradeoff.



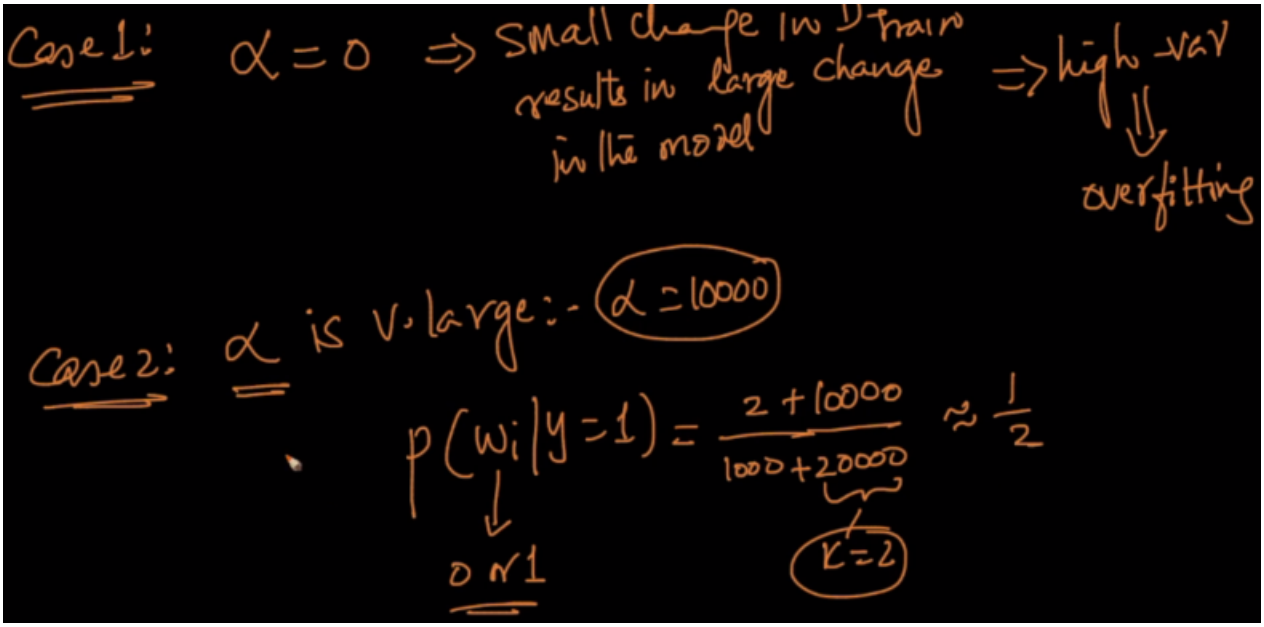

Case 1: when alpha = 0
When alpha = 0, then we will overfit the data because for the words that occur very less frequent also, we calculate the probability value.



By removing the two texts out of all the words, the probabiliy of occurance will become zero.



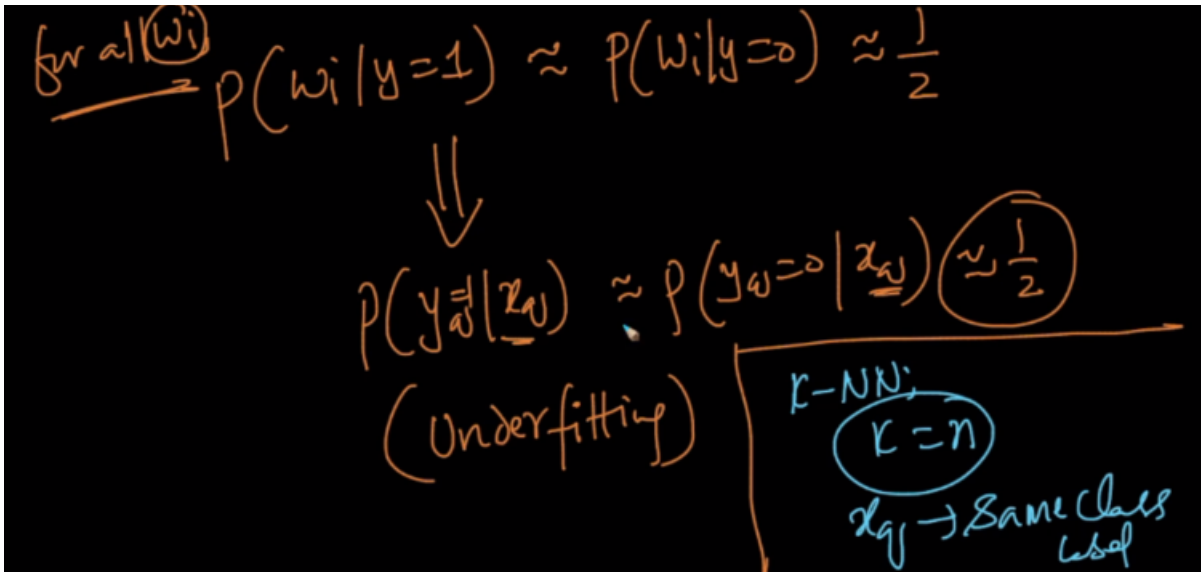

Case – 2 : alpha is very large.



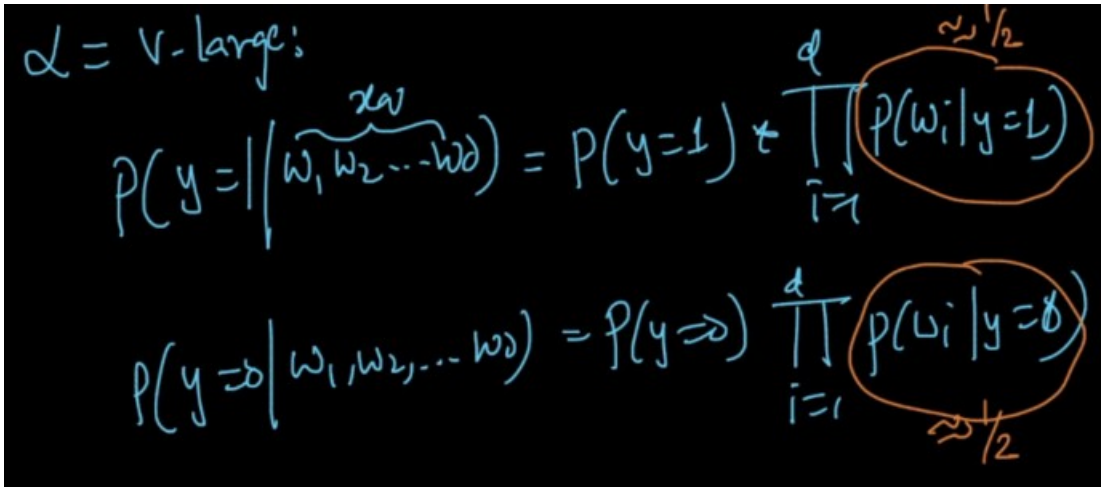

If alpha is very large then the every word in the text will move towards the uniform distribution, i.e, 1/2.

Given the model, that cannot distinguish between the two classes for large values of alpha.

This is the case of under fitting.

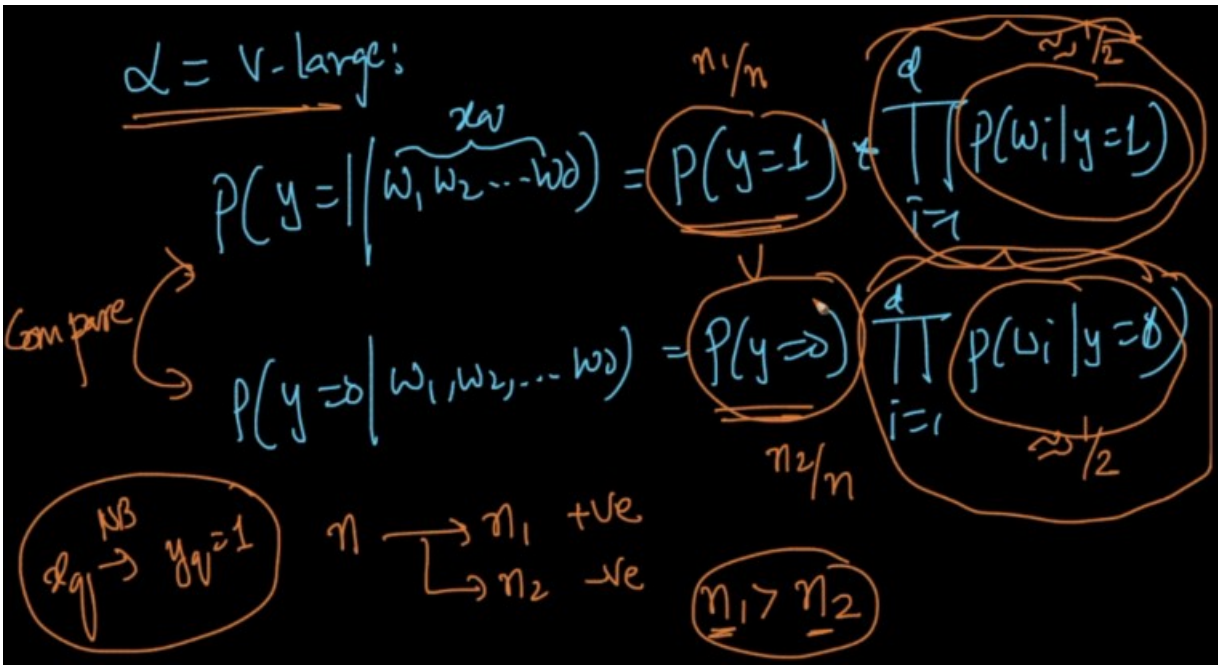



Comparision of NB with KNN:

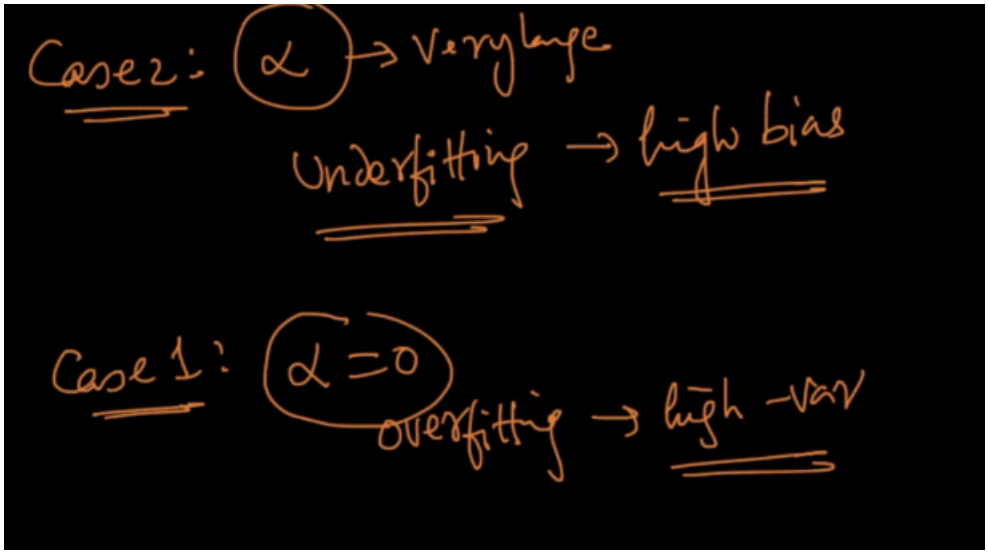



When alpha is very large then the probability of the majority class wins over the minor class. In addition if we have high alpha values then every probability values of the words will be equal to half.

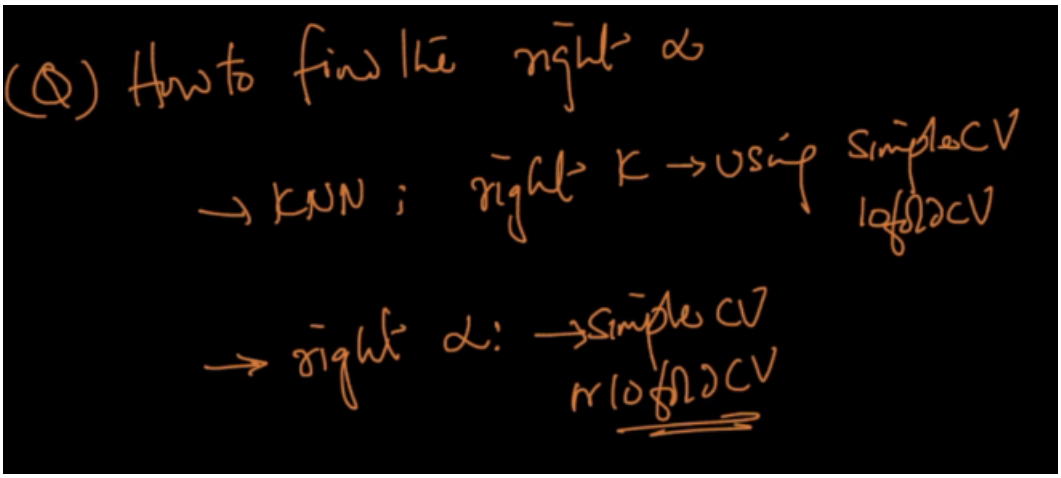
This is same as KNN with K is the hyper parameter.



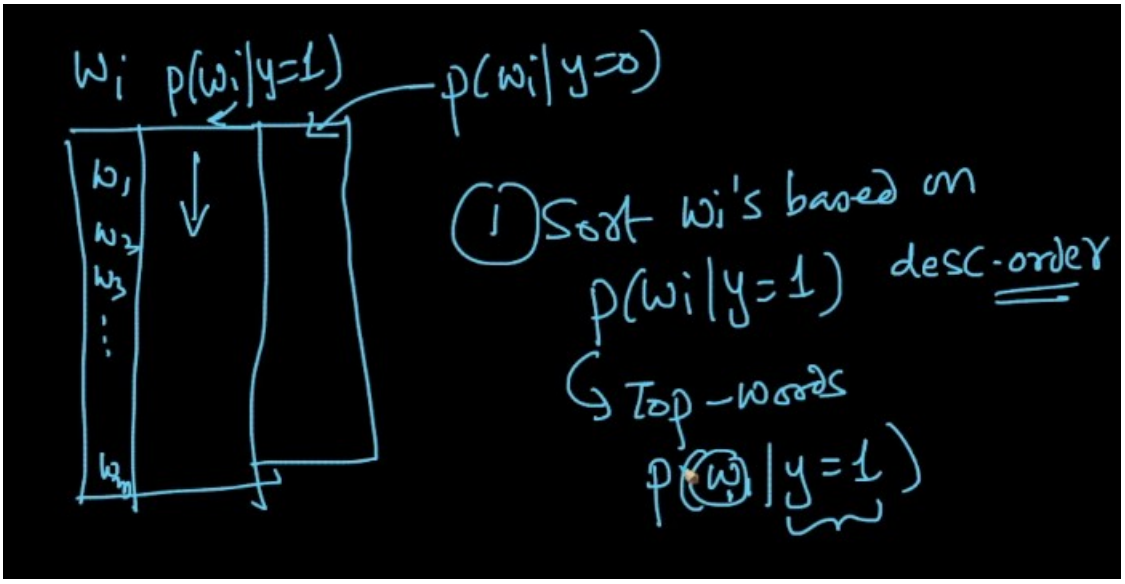

Summary:




How to find the right alpha ?

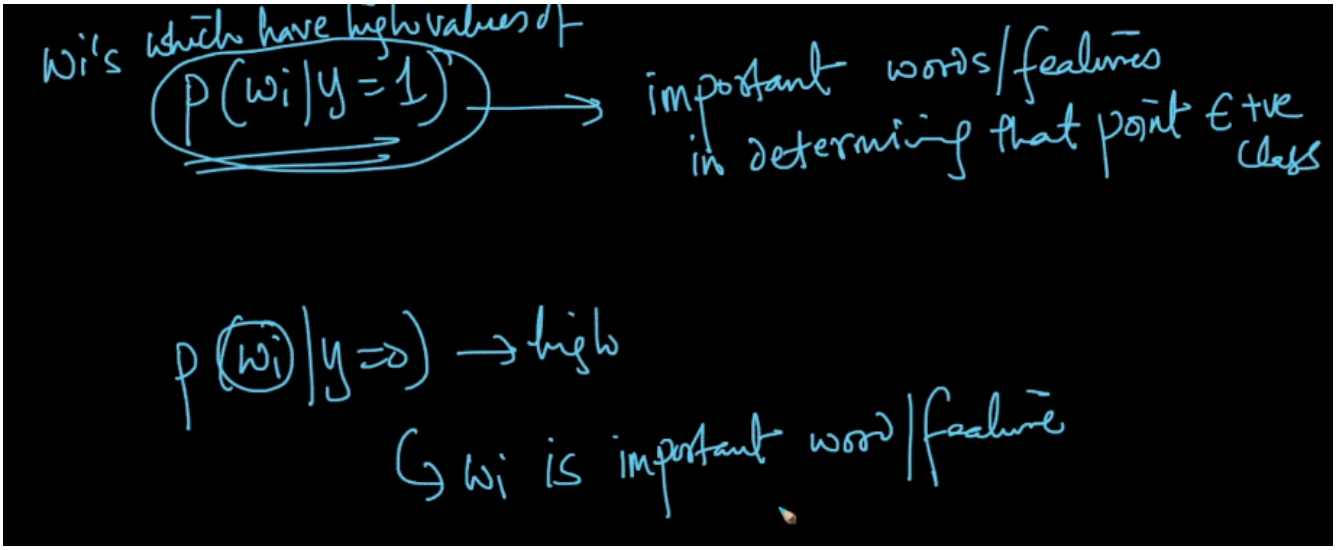



Feature importance and iterpretability:
For every word w1 and probability of likelihood for the both classes is calculated.



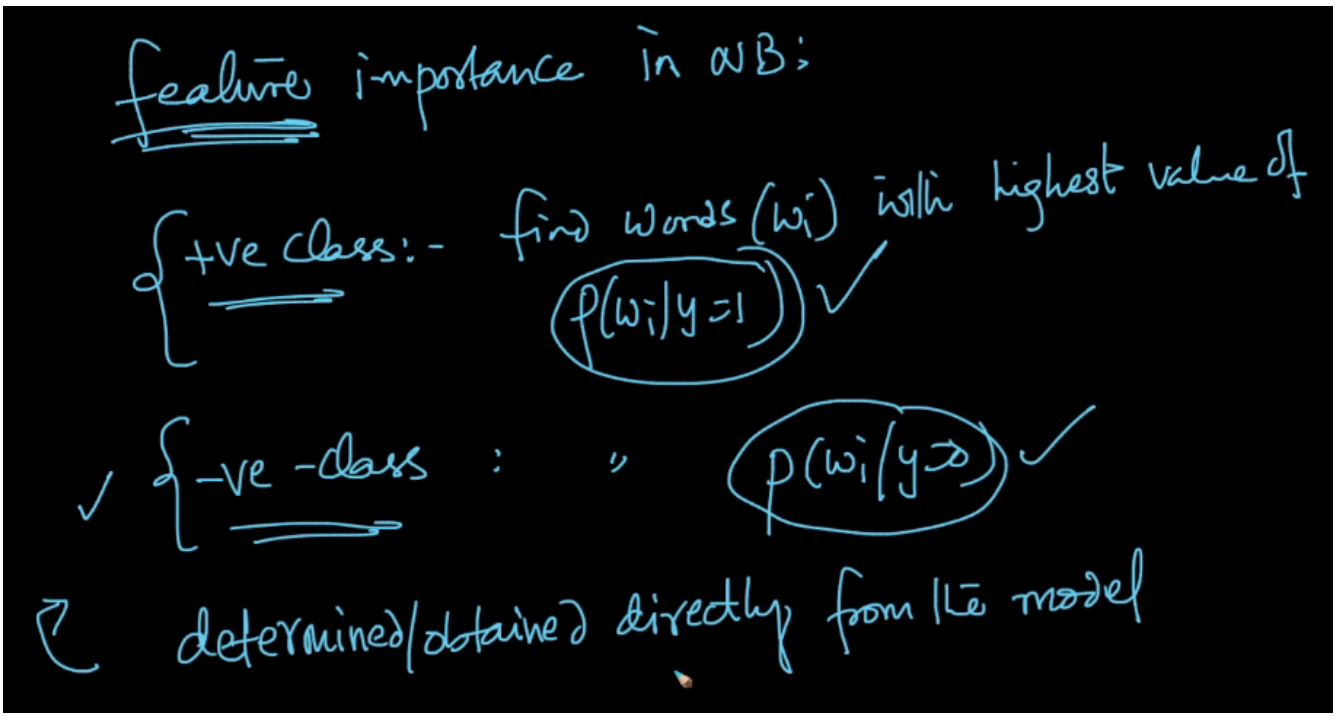

The words which have the highest probability are very useful or important features or words in determining the data point belongs to that class.

Those words are very important word or feature that a point is more important.




Feature importance in naive bayes is trivial, for a class we can figure out the words with highest value of probability.

Feature importance can be obtained deirectly from the model.



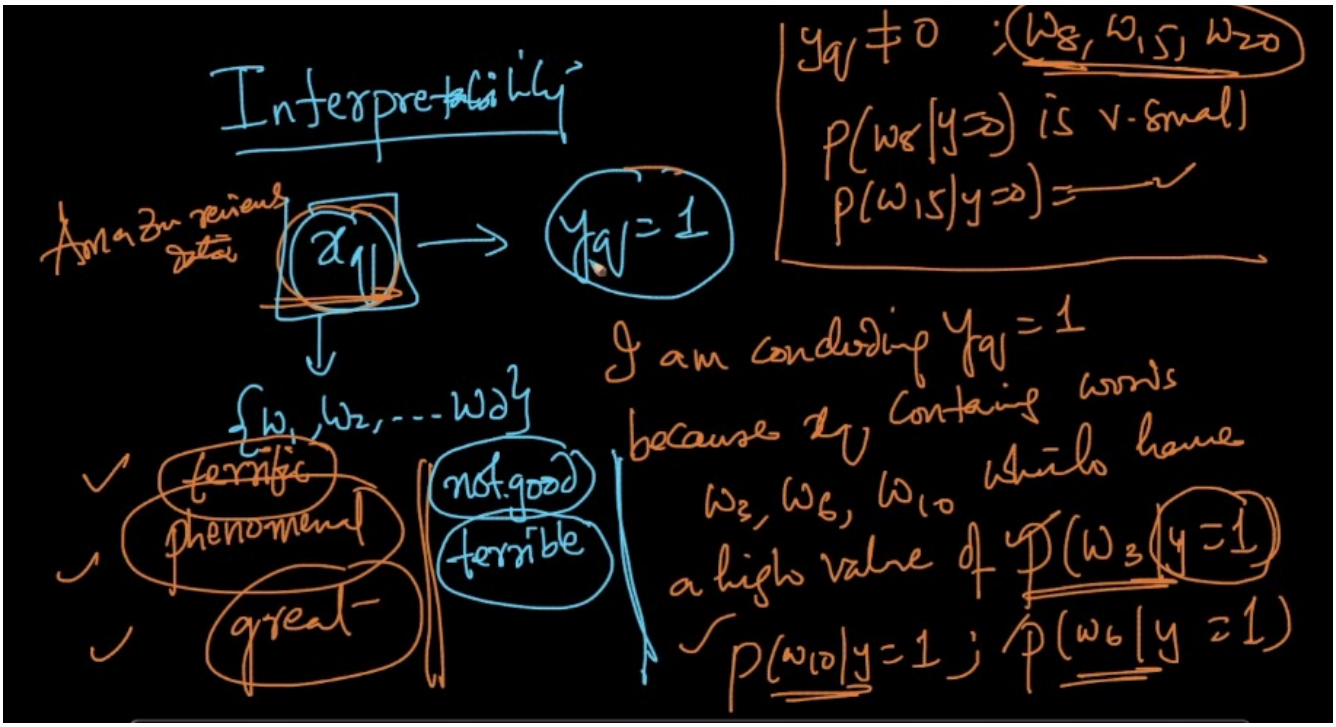

In KNN it is not possible to get feature importance, we used forward feature selection as the criteria.

Interpretability:

If the query point has the highest probable features, then we can conclude the query point belongs to the class, because the query point has the high probable words.

We can delare the query point as a positive or negative, if it contains the words that has the most probable value words of the classes.like(for pos – great, good, terrific, phenomenal,  neg – not good, terrible).

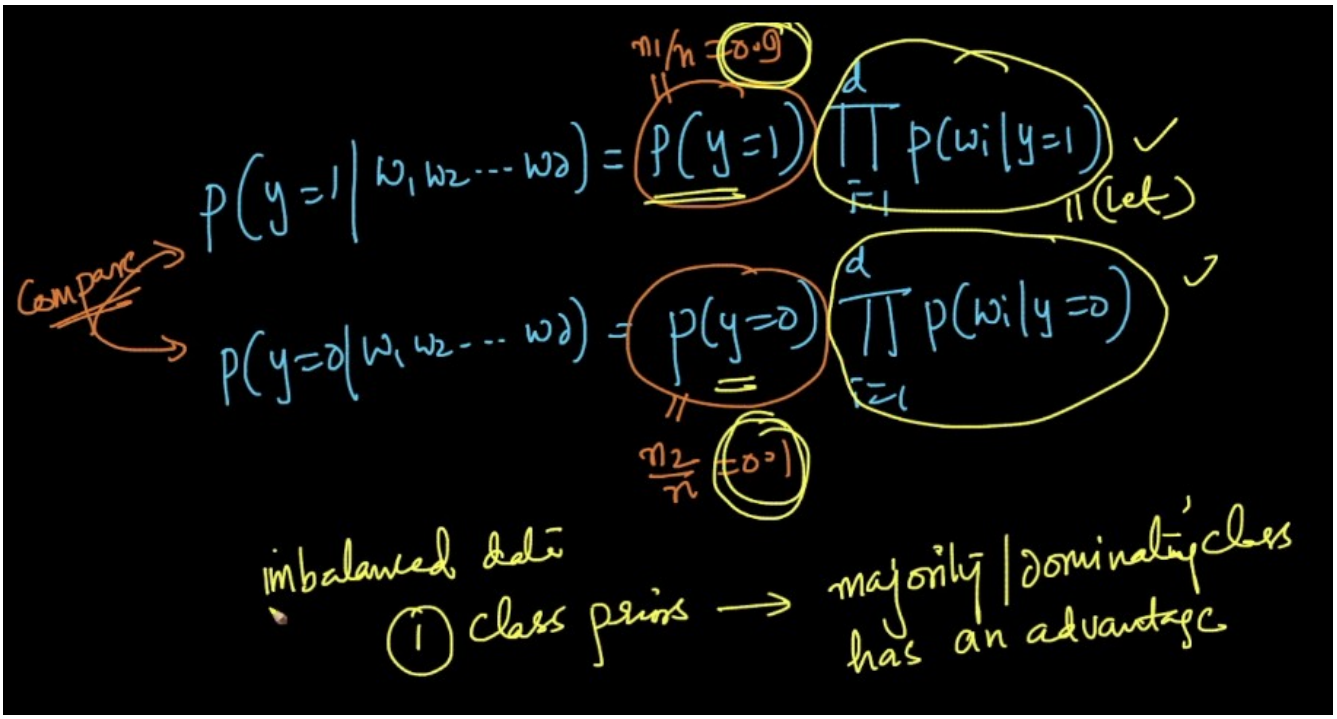
This is how we interpret the queried review with the probability values of the traning data.



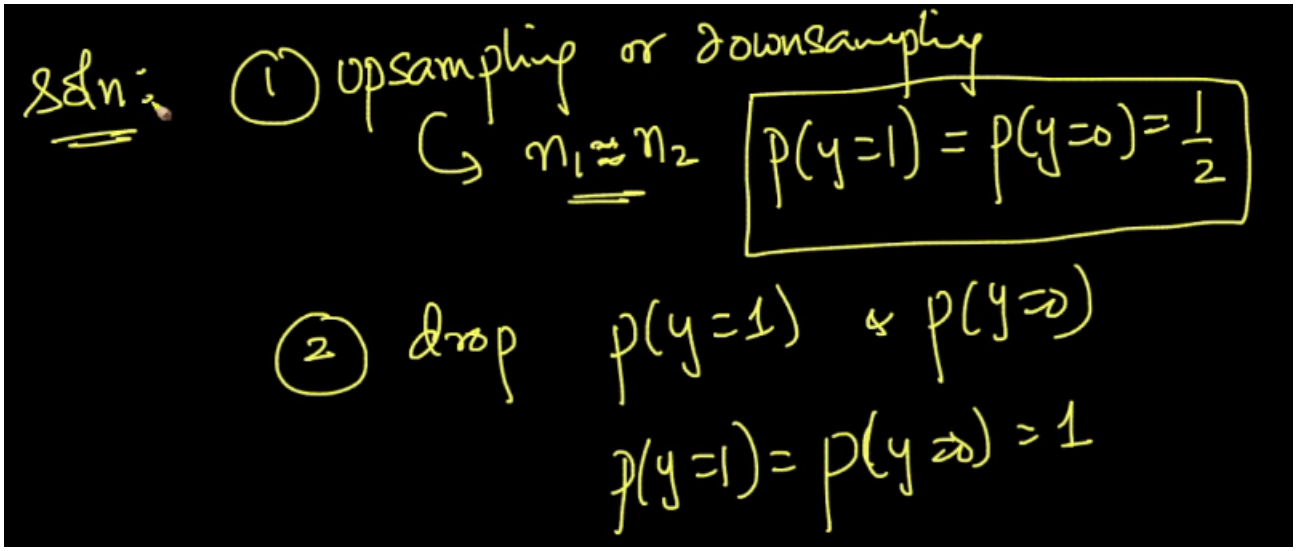
The conditional probabilities are nothing but the likelihood values.

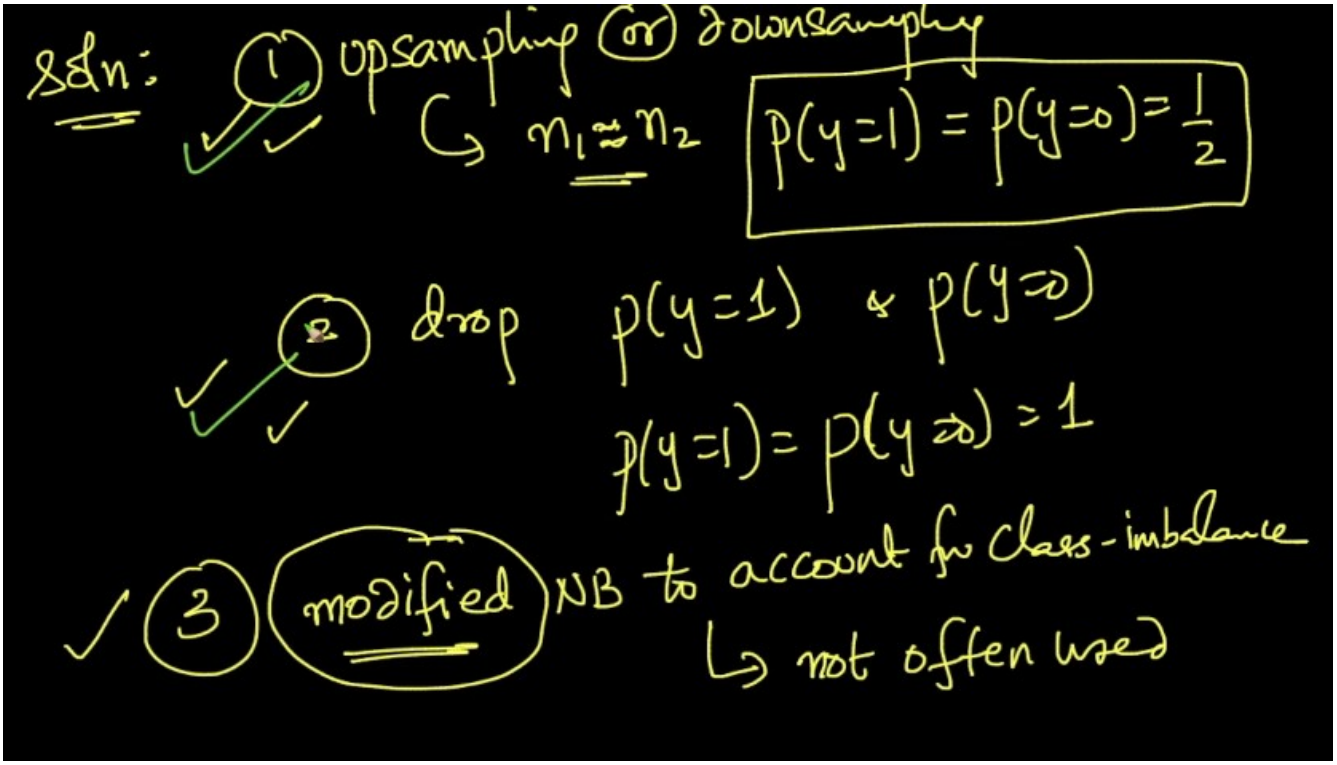
Imbalanced data:

In case of imbalanced data, the majority or dominating class has a advantage. Though the conditional probabilities are same (or) equal.



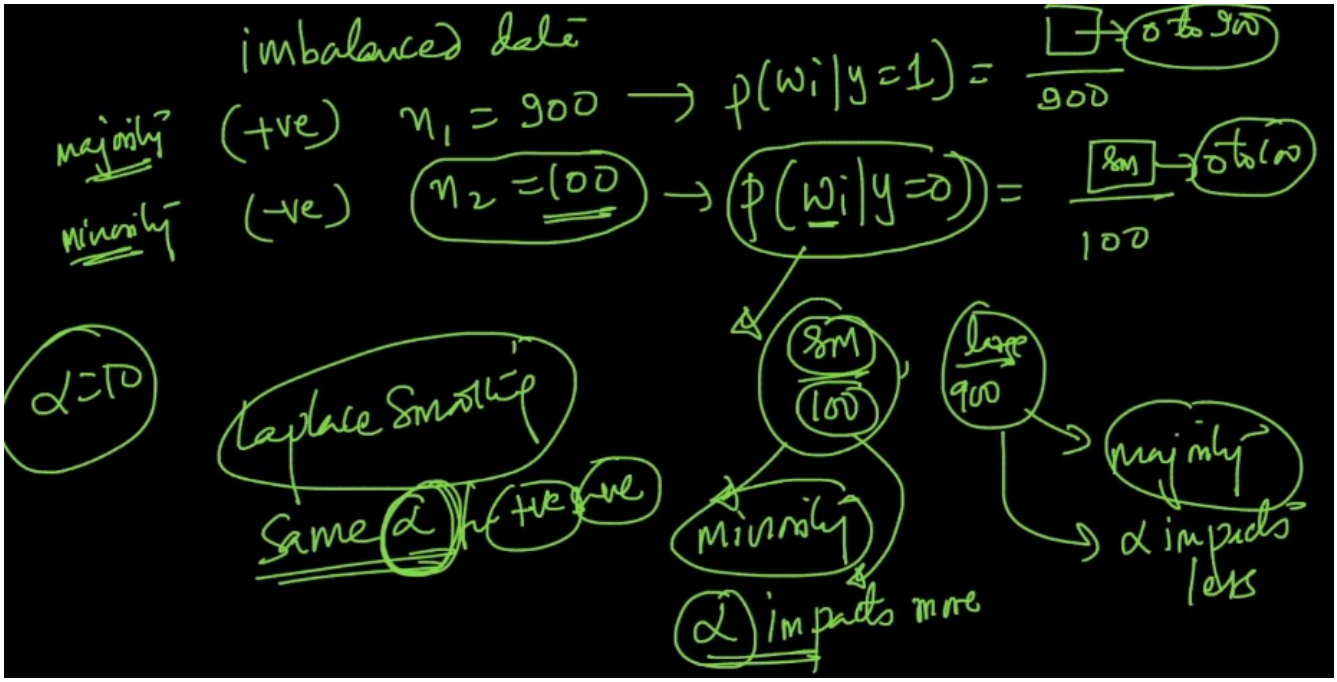

We can use the standard techniques of upsampling and down sampling. (or)
**we can drop the prior probabilities.**



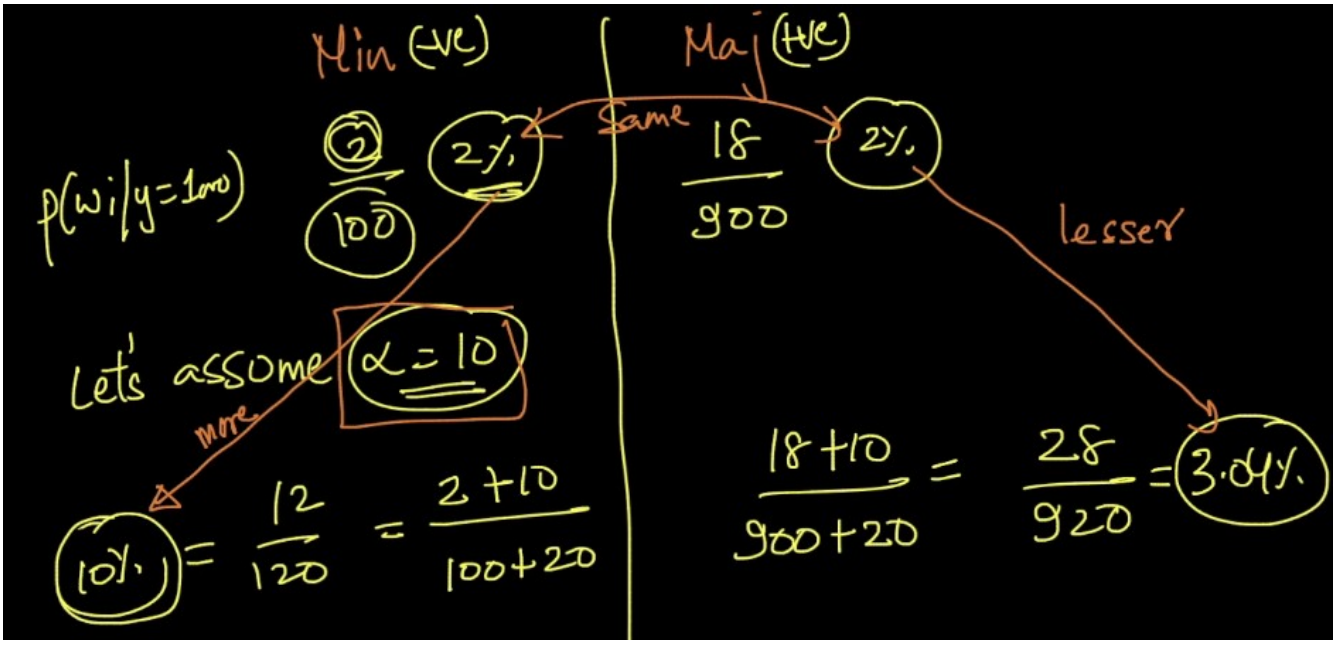



The majority class has the more range of values than the minority class range of values.

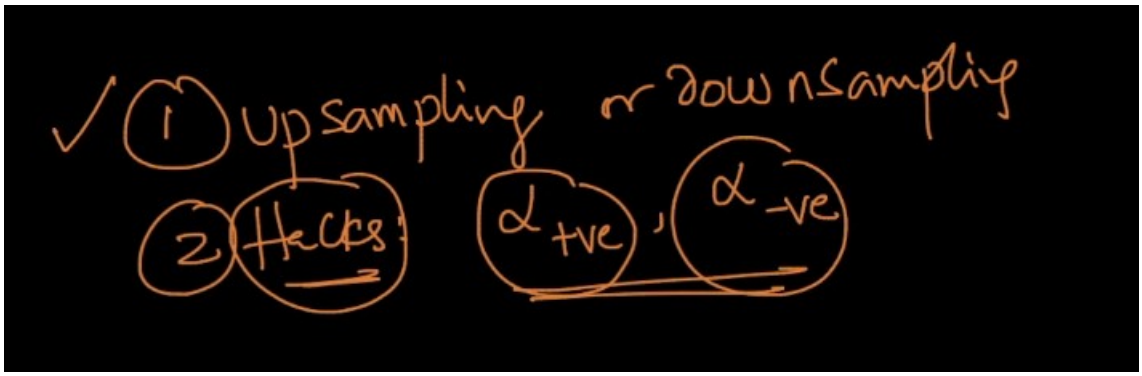
Impact of the laplace smoothing:



With the laplace smoothing, the miniroty changes a lot than the majority class. For the same alpha value.



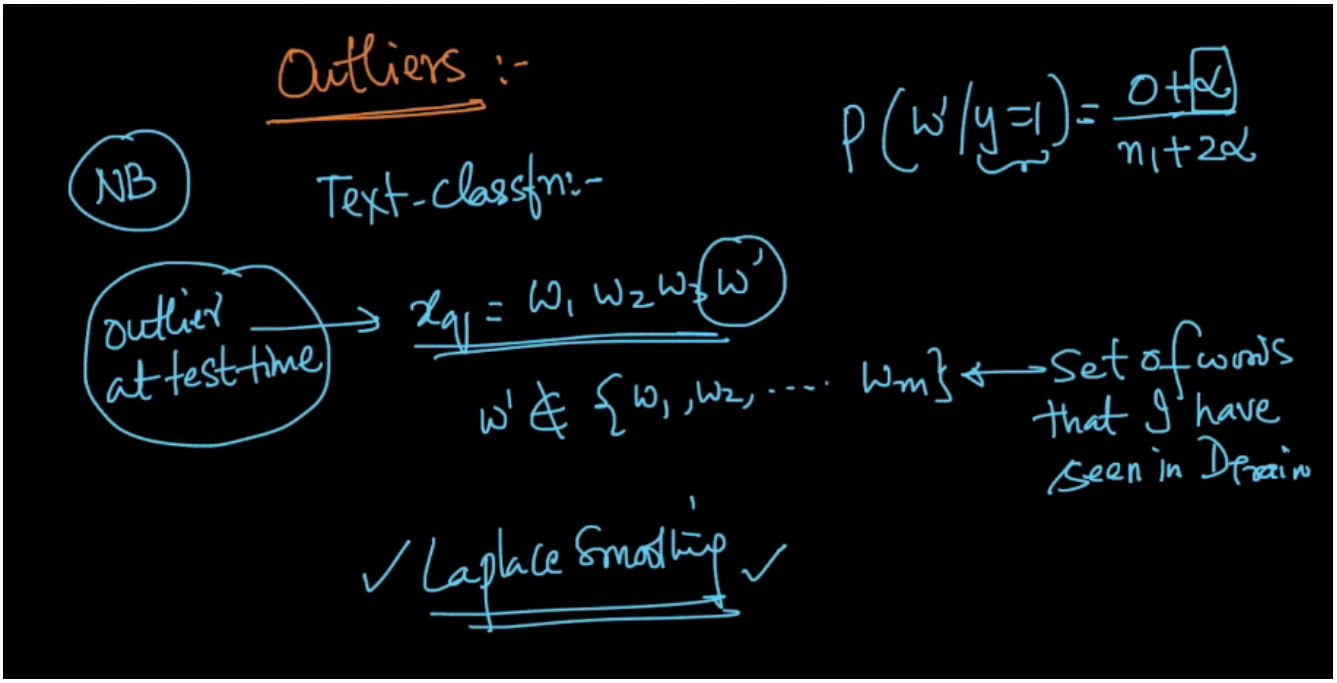

Upsampling and down sampling can be used to make the affect of alpha less. We can use different alpha values for positive and negative classes.(this is a hack).



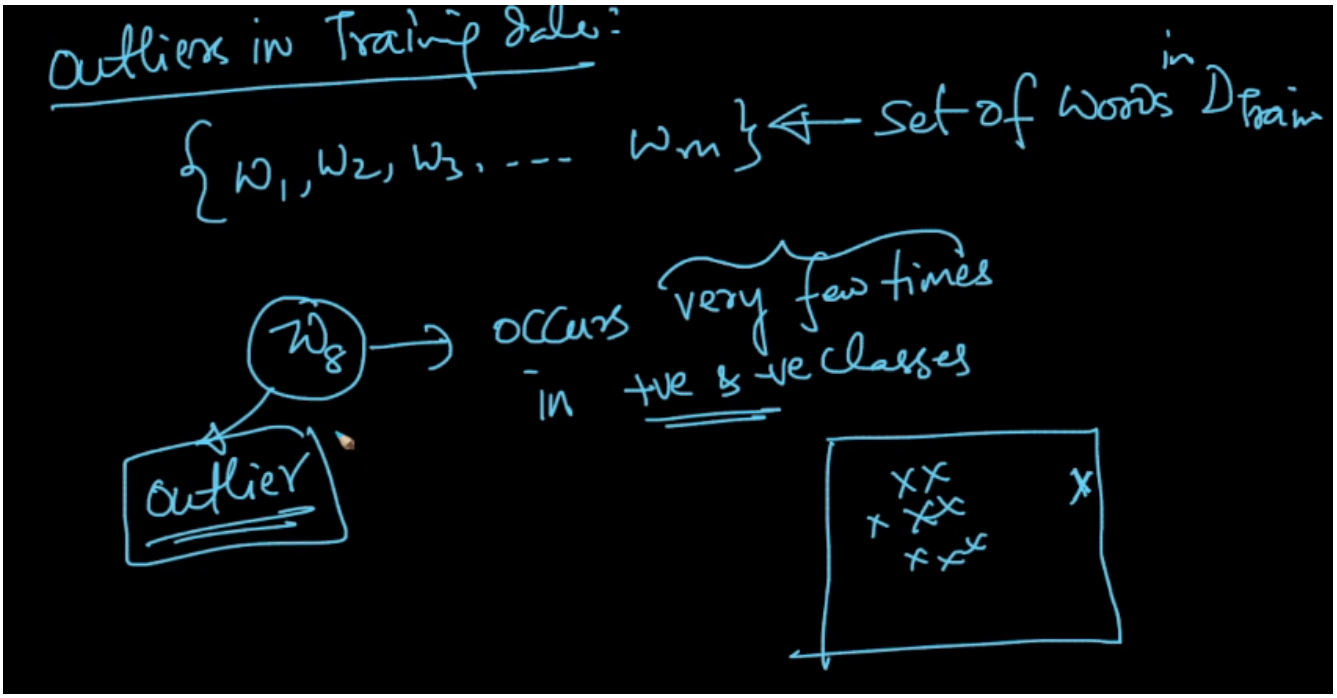

Naive bayes imacted by imbalanced data, it affects the prior probabilities and also impacts the likelihood probabilities for the same values of alpha.

Outliers:

During test time:
If we get the outlier during the test time, laplace smoothing takes care of the outlier point.

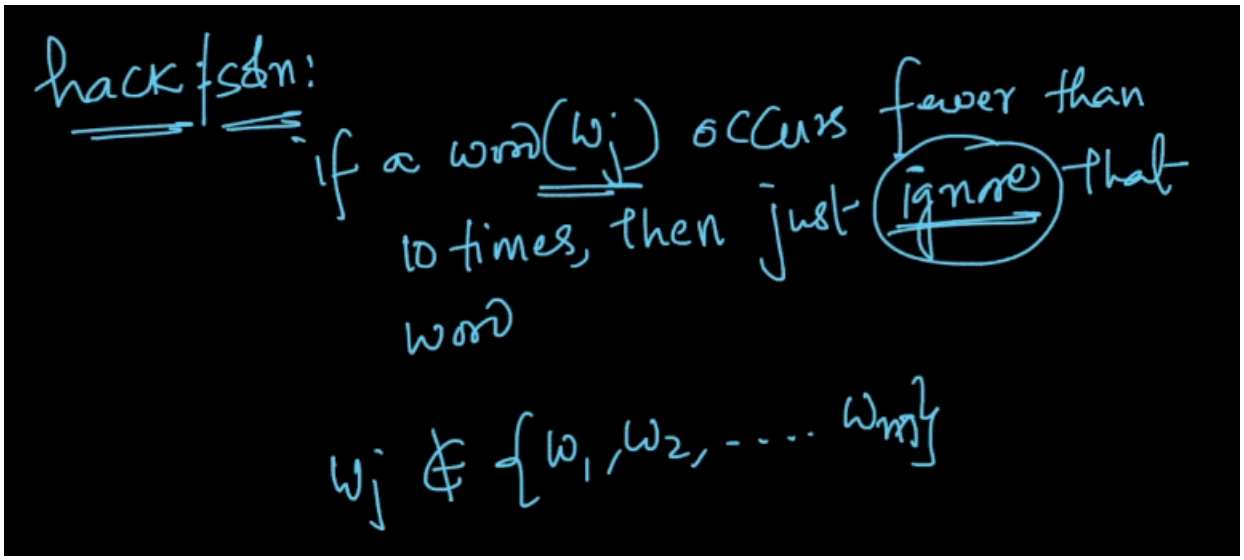



During outliers in traning data:
If the words in the positive and negative classes occur very few times then this can be taught of an outlier.
Outlier can be taught of there are not many points around the point. This can be taught of an outlier.



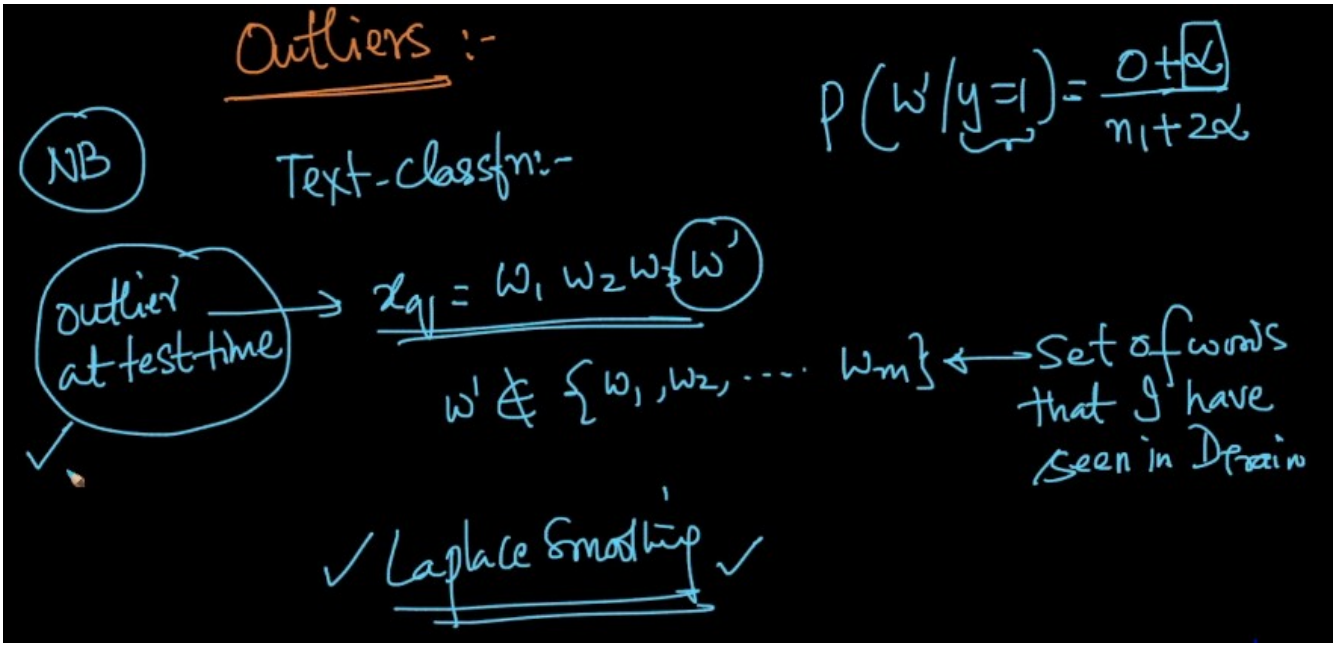
First solution:

If a word occurs fewer than 10 times, then just ignore that word. Droping that one word can be the choive of doing.
Even if occurs in the traning data.




Second solution:

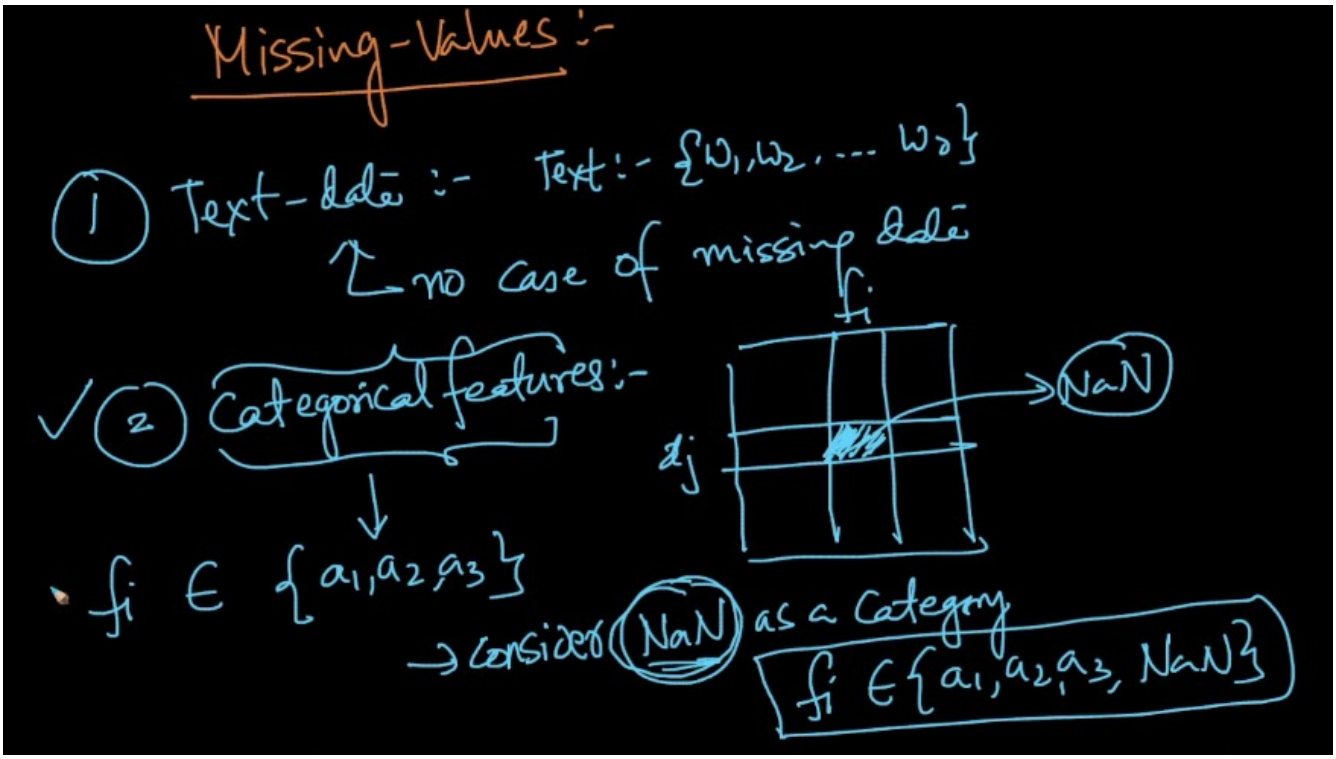
The solution is using the laplace smoothing.



Missing values:

In the case of text data, then there is no case of missing data.

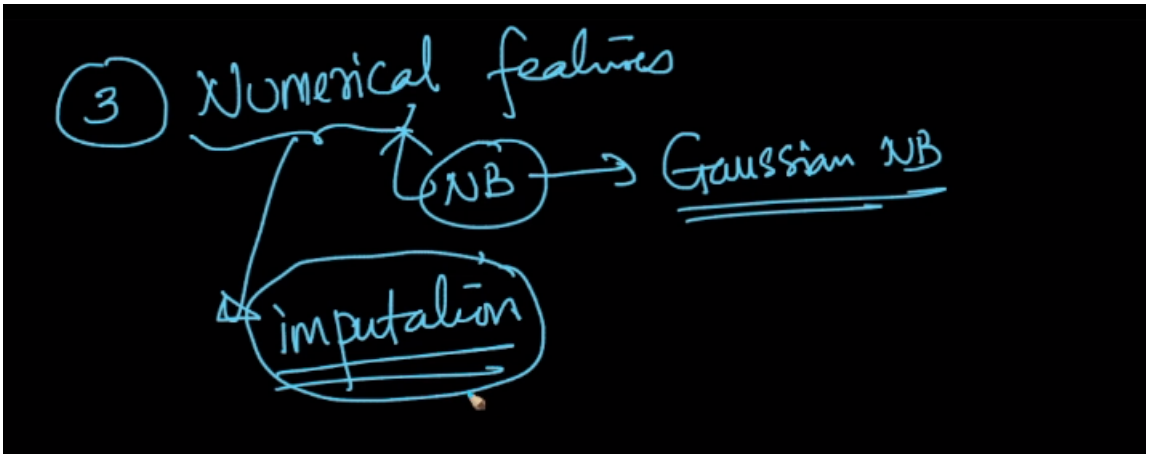
In case of categorical data, we can use the new feature called NaN and make that as the new feature.




For numerical features:

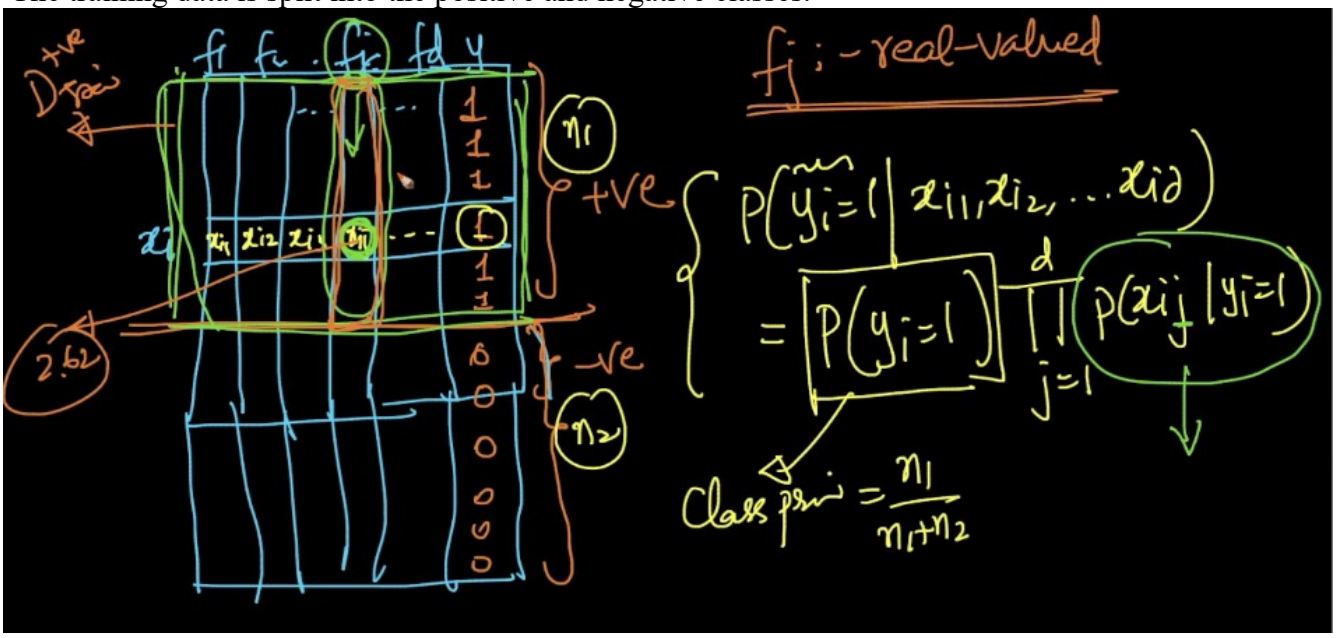
We use the standard imputaions techniques for the missing values.

We use GaussianNB for numerical features.



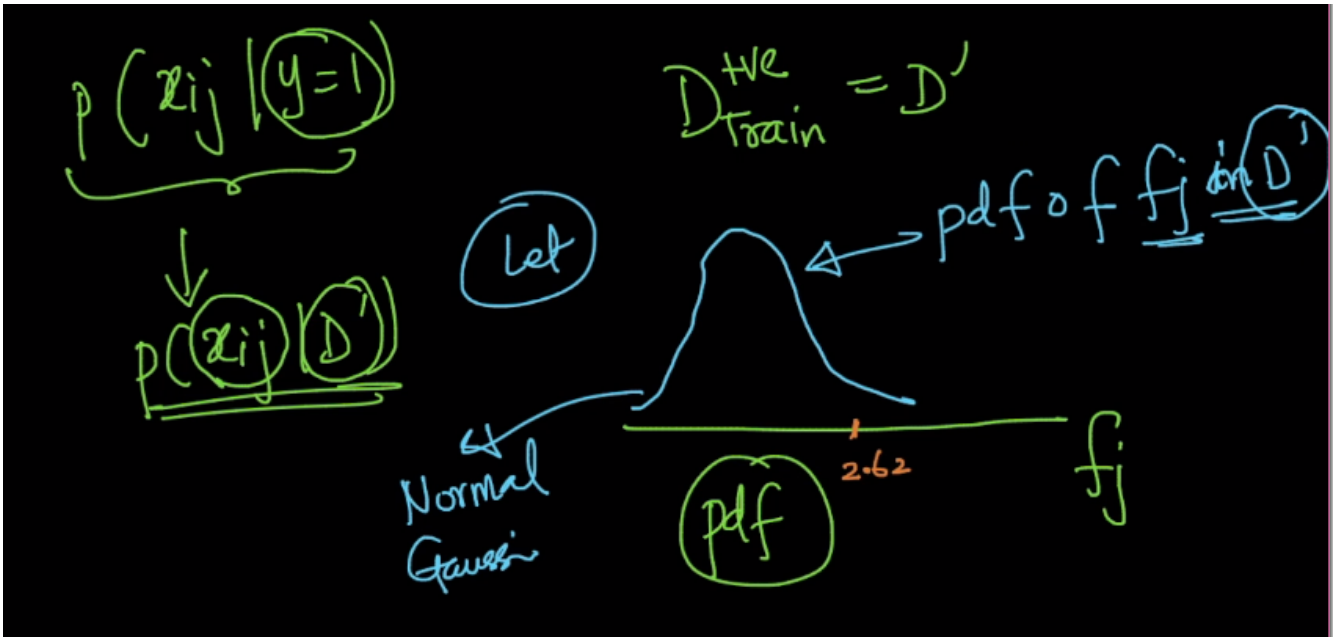
Handling Numerical features(Gaussian NB):

Step 1:
The training data is split into the positive and negative classes.

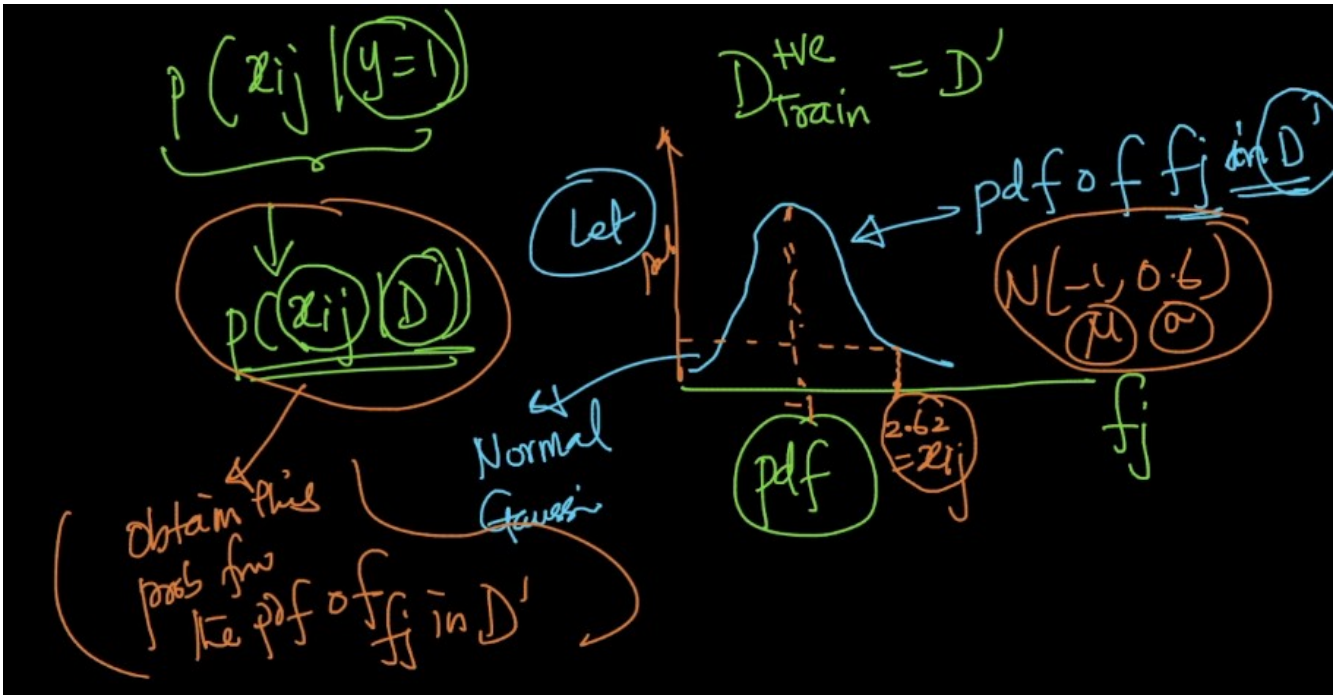



Calculating the probability for the two classes. The pdf is only plotted for only the positive labelled dataset (or class representative)

The pdf of the feature is plotted.

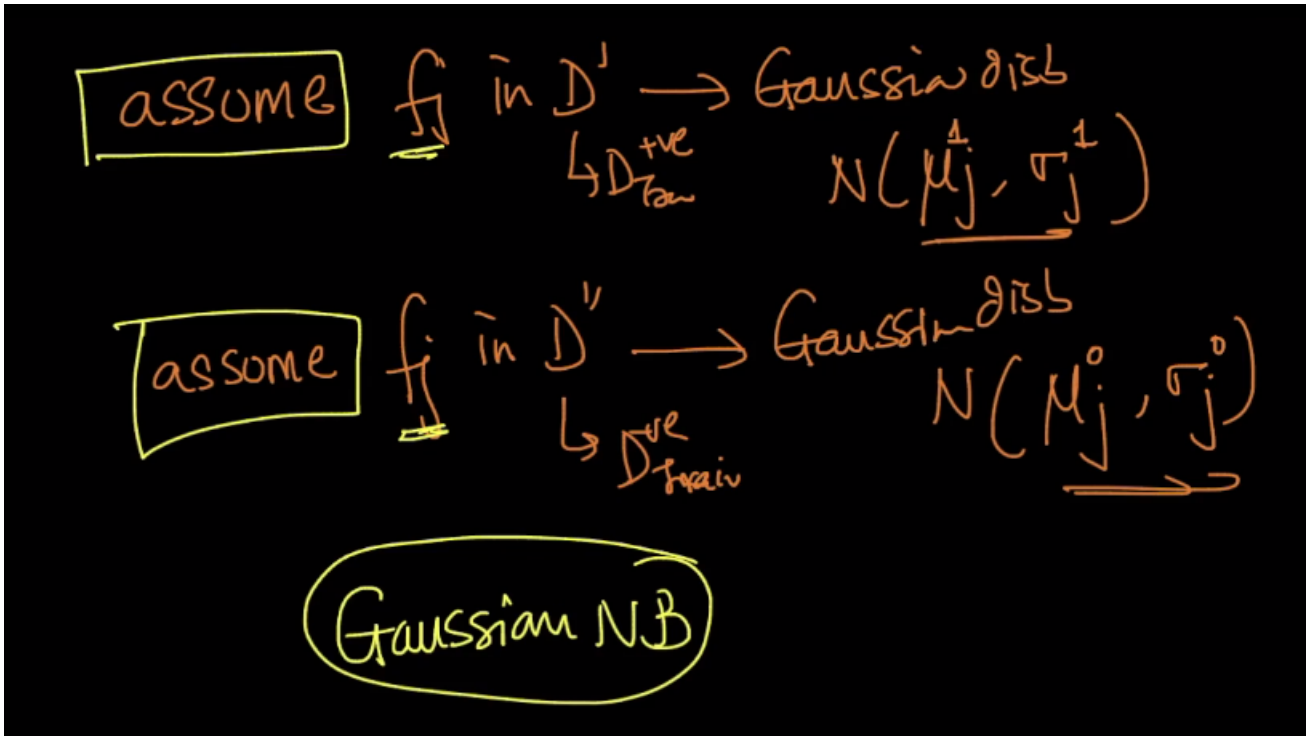


We can obtain the probability of single value from the feature by plotting the pdf of the feature.




Similarly we can get for the other class also.

It is often assumed that the distribution of the feature is alwayas a gaussian.



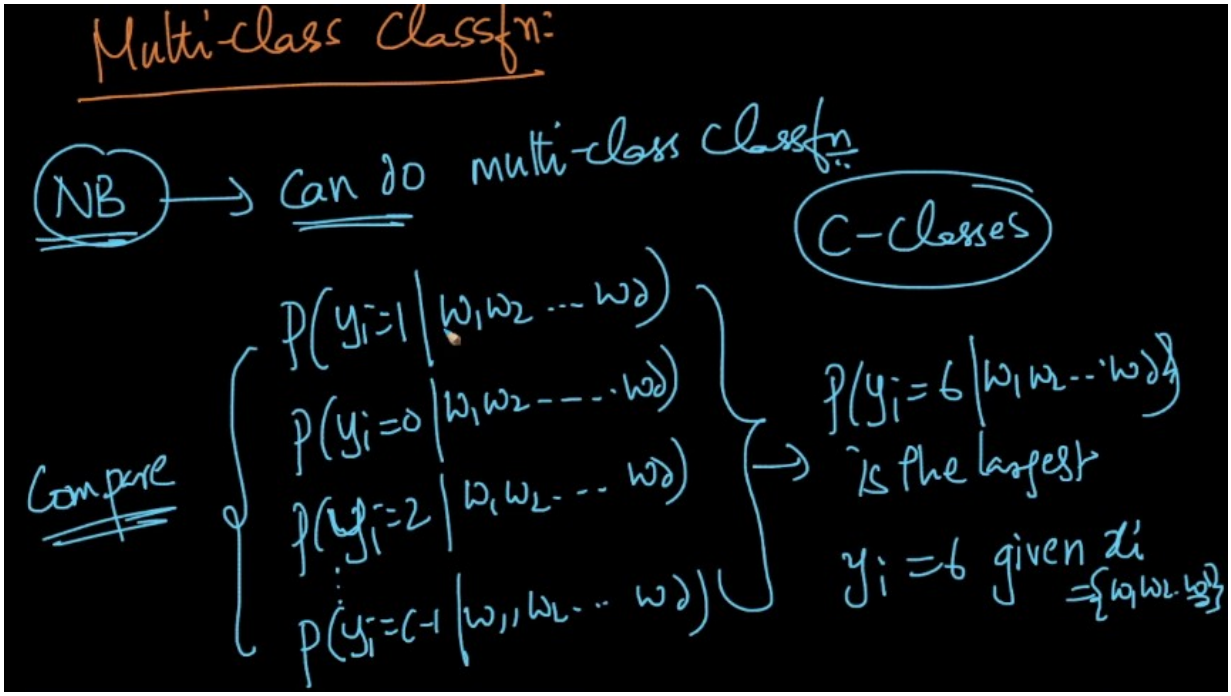
Such a model is called the gaussian naive bayes.

There are different types of NB such as:

- BernoullliNB (it is a binary rarndom variable)
- MultinomialNB.
- GaussianNB.


**Each of the feature is conditionally independent in case of Naive Bayes.**
Multiclass classification:

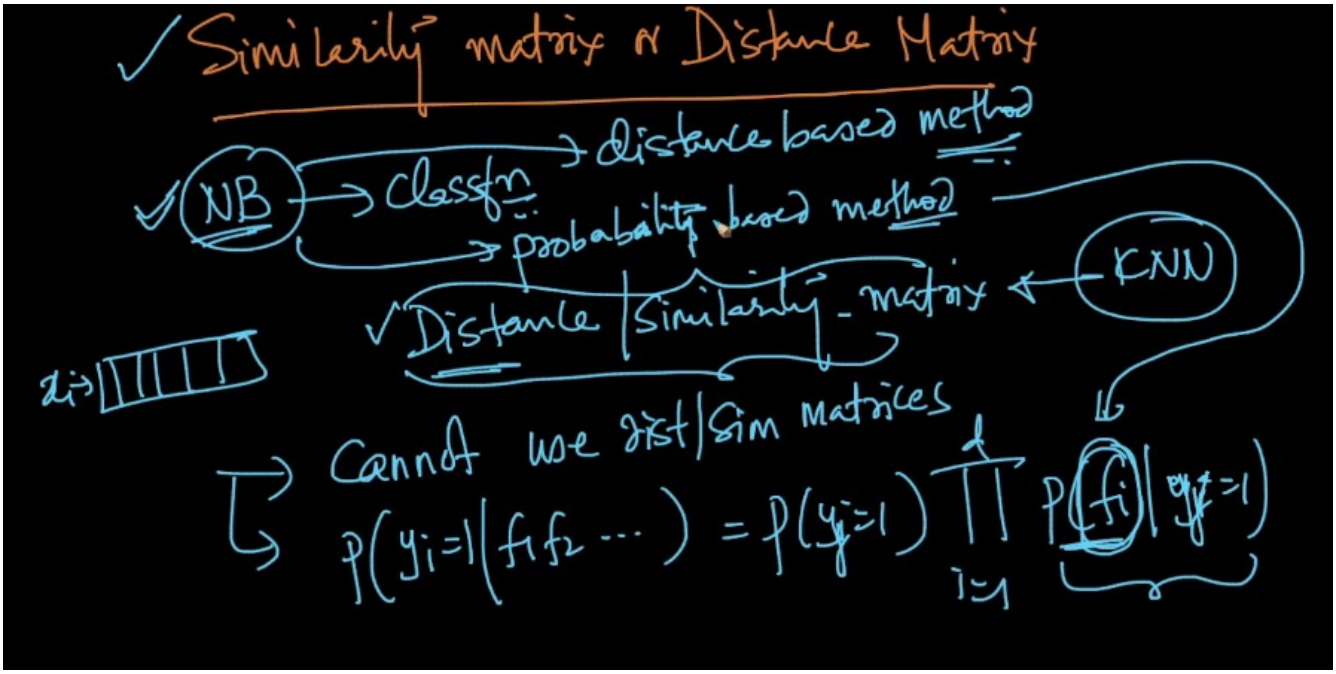
We can compute priors and likelihood probabilities for any number of classes in the dataset.Then compare all of them.




Similarity matrix or a distance matrix:

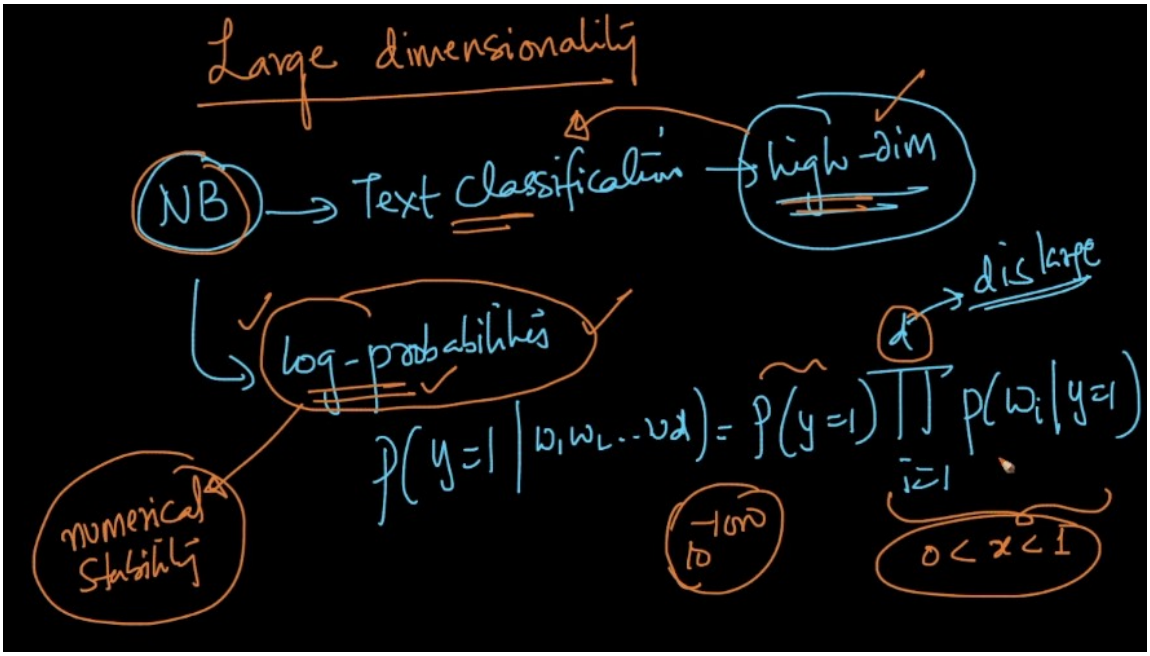
In general NB cannot use the distance or similarity matrix. We cannot compute the likelihood prbabilities.

NB does not the distance based algorithm. It is the probability based method.
We need actual feature values for calculations for probabilities.



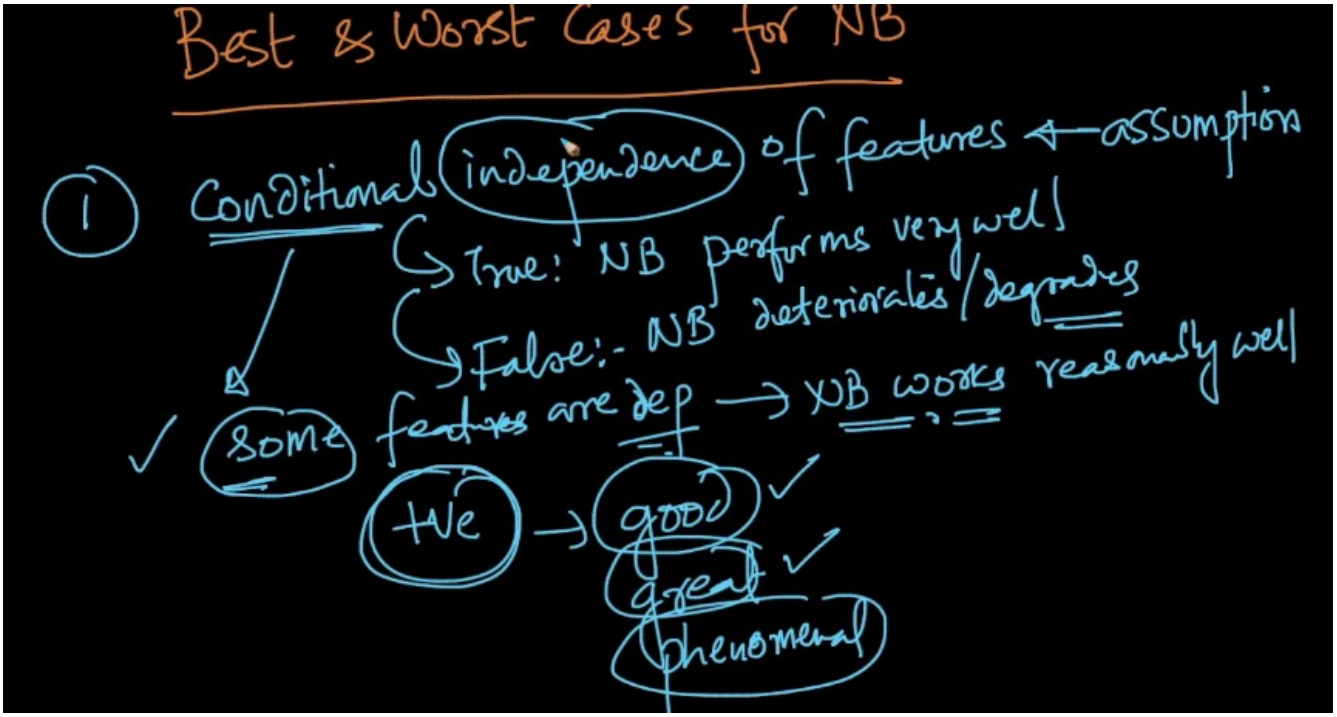

Large dimensional values:
In case of large dimensions the calculation of likelihood probabilities multiplication can cause the

numberical stability issues. Use the **log probabilities** to overcome this problem in high dimensional data.

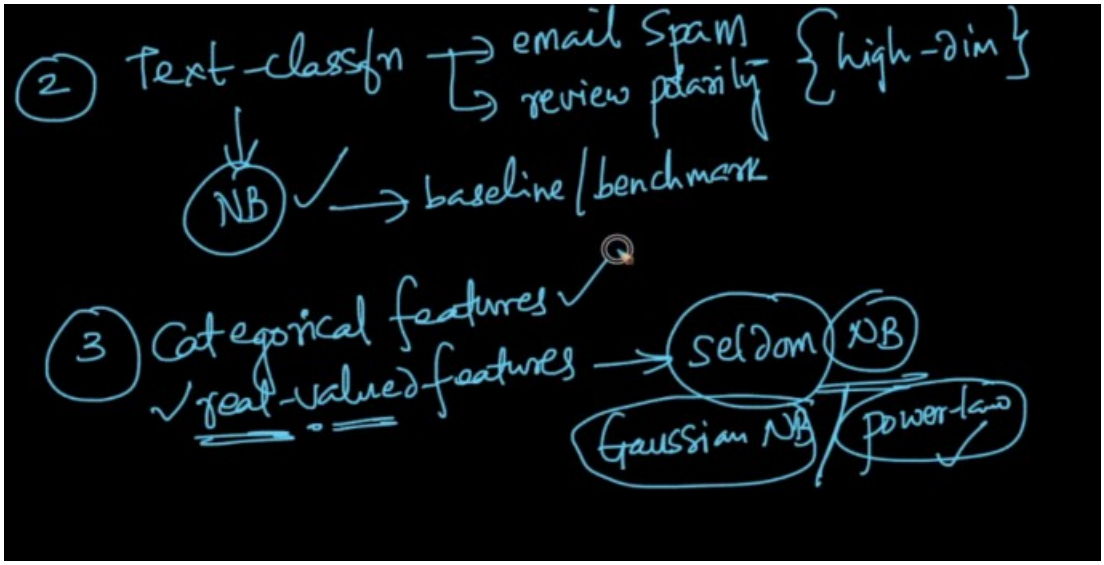



Best and worst cases of NB:

NB assumes conditional independence of the features assupmtion.
In some cases the words in text data does not mean too different, they mean one and the same like the words good, great and phenomenal.



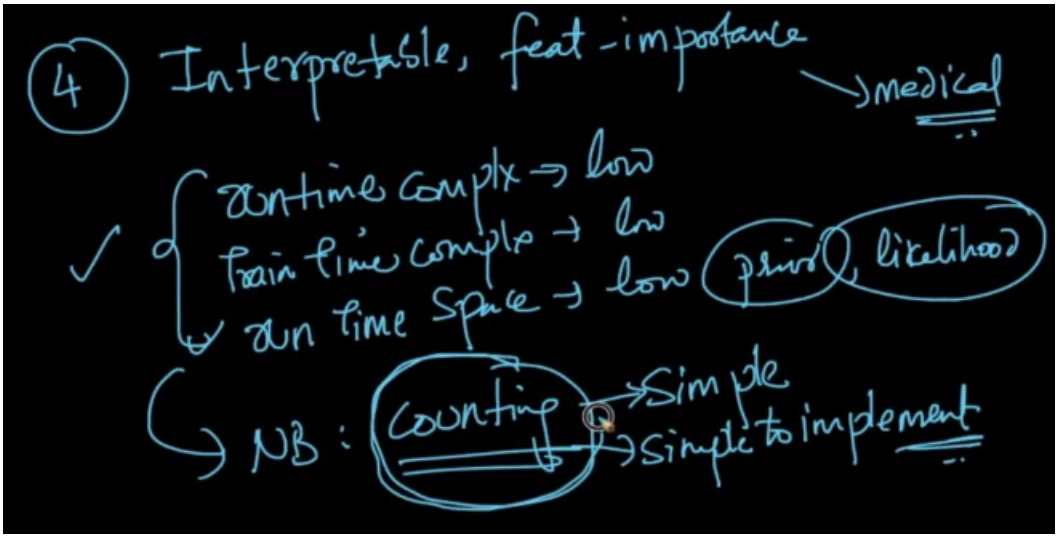

Even some features are dependent then naive bayes works reasonably well.
Theorotically, naive bayes works only when the features are independent.




Naive bayes is not most often used in case of real – valued features.

Naive bayes is super interpretable.

The run – time complexiety is also low.




Naive bayes can easily be overfit the data. If there is not laplace smoothing.