Mathematical formulation

data → x_1 x_2 ..... x_n Label (k no of classes)

Query pt → $< x_1 x_2 x_3 \ldots x_n >$

- Since, there are k no of classes, Naive Bayes will find k no of probabilities, one for each class for new query point

↳
- Let's call this query pt X^T collectively
- Since there are k no of classes, let's say $(y_1\ y_2\ y_3 \ldots y_k)$, Naive Bayes will find probabilities for each class
- If there were 2 classes, ex Yes or No this will be called Binary
- Since there are k no of classes, the probab will be

→ $P(y_1 | X_T) = P(X_T | y_1)\ P(y_1) / P(X^T)$

→ $P(y_2 | X_T) = P(X_T | y_2)\ P(y_2) / P(X^T)$

→ $(P y_3 | X_T) = P(X_T | y_3)\ P(y_3) / P(X_T^T)$

:

→ $P(y_k | X_T) = P(X_T | y_k)\ P(y_k) / P(X_T$

- We can remove denominator, for simplification
- In among class whose probabilities is most, that class will be assigned to our query X_T

- Let's focus on general case
- $P(Y_k \mid X_T) = P(X_T \mid Y_k)\, P(Y_k)$
- putting back X_T, and our equation will be
- $P(Y_k \mid X_T) = P(X_1 \cap X_2 \cap X_3 \cap \dots \cap X_n \mid Y_k)\, P(Y_k)$
- Intersection is there as in query it represents events are happening at same time that's how query was made
- $\cap$ can be represented as , as both are same

⇒ can be rewrite as

- $P(Y_k \mid X_T) = P(X_1, X_2, X_3, \dots X_n \mid Y_k)\, P(Y_k)$

We know $P(A \mid B) = \dfrac{P(A, B)}{P(B)}$

- Therefore ⇒ $P(Y_k \mid X_T) = \dfrac{P(X_1 X_2 X_3 \dots X_n, Y_k)}{P(Y_k)}\, P(Y_k)$

$P(Y_k X_T) = P(X_1 X_2 X_3 \dots X_n, Y_k)$

We know ⇒ $P(A \mid B) = \dfrac{P(A, B)}{P(B)}$

⇒ $P(A, B) = P(A \mid B) \times P(B)$

We have $P(Y_k X_T) = P(\underbrace{X_1}_{A}, \underbrace{X_2, X_3, \dots X_n, Y_k}_{B})$

Considering $A \to X_1$ & $B \to X_2, X_3 \dots X_n, Y_k$ apply

underline rule ⇒ $P(A,B) = P(A|B) \times P(B)$

Hence,

$\Rightarrow P(Y_K X_4) = P(X_1 | X_2, X_3 \cdots X_n, Y_K) \times \underbrace{P(X_2, X_3 \cdots X_n Y_K)}_{2^{nd}}$

Appling same rule again on 2^n term

$= P(X_1 | X_2, X_3, \cdots X_n, Y_K) \times P(X_2 | X_3, X_4 \cdots X_n, Y_K) \times \underbrace{P(X_3 X_5 \cdots X_n, Y_K)}_{3}$

⇒ Applying same again on 3rd term

$= P(X_1 | X_2, X_3 \cdots X_n, Y_K) P(X_2 | X_3, X_4 \cdots X_n, Y_K) \times P(X_3 | X_4, X_5 \cdots X_n, Y_K)$

⇒ Applying till last

$\Rightarrow (P(X_1 | X_2, X_3, \cdots X_n, Y_K) \times P(X_2 | X_3, X_4, \cdots X_n, Y_K) \cdots \cdots P(X_{n-1} | X_n, Y_K) \times P(X_n | Y_K) \times P(Y_K)$ → main eqⁿ

Main cheez: Naive Bayes takes Naive assumption, that in your data ther are $X_1 X_2 \cdots X_n$ columns, and you take assumption that all the features/columns are independent of each other

- That means X_1 doesn't depend on $X_2 X_3 X_4 \cdots X_n$
- same goes for X_2 " " on $X_3 X_4 \cdots X_n$

→ • We have this equation
• let's take one term out of it just for explaination

- $P(X_1 | X_2, X_3 \ldots\ldots X_n, Y_k)$ (remember; represents n)
- We know that x_1 doesn't depend on $X_2 X_3 \ldots X_n$, but depend on Y_k
- We know that when events are independent :- $P(A|B) = P(A)$
- But in our case, X_1 ~~depend~~ independent of $X_2 X_3 \ldots X_n$ but depend on Y_k
- Something like this → $P(A | B \cap C) = P(A | B, C)$
- where A is independent of B, but depends on C
- In this situation, you can write
- $P(A|B,C)$ as $\Rightarrow P(A|C)$
- because of $P(A|B) = P(A)$ when A & B are independent
- Apply same logic to main eqn
- Our eqn will be modified to :-

$$P(X_1 | Y_k)\ P(X_2 | X_k)\ P(X_3 | Y_k) \ldots\ldots \ldots\ldots\ldots P(X_{n-1} | k)\ P(X_k)$$

Ex for class $y=1$, the prob will be :

$$P(X_1 | X_1) \times P(X_2 | Y_1) \times P(X_3 | Y_1) \ldots\ldots P(X_1)$$

- Similarly, you will find all class probabilities
- Compare each class Prob, whoever is highest that will be assigned that label

- That's why the eqn was breakdown into simpler terms (in intuition part (This was the proof)

* Overview of ~~training~~ training and testing phase of Naive Bayes

Training Phase:

- In training phase you calculate all possible probabilities

Ex - In a dataset we used, we have two classes → 'Yes' and 'No'

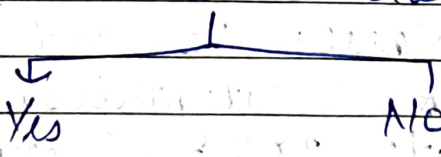
- for every input column, you check how many categories you have
- Ex In ~~Output~~ Outlook col we had categories :- Sunny, Overcast, rainy
- we have two classes Yes and No, and in Outlook → 3 categories
- We will calculate 3 x 2 = 6 probabilities
- i.e P(Sunny | Yes)
  P(Sunny | No)
  P(Overcast | Yes)
  P(Overcast | No)
  P(Rainy | Yes)
  P(Rainy | No)

- Same goes for other columns.
- Temperature contains Hot, Mild cool.
- 6 conditional prob will be calculated
  - ie → P(Hot | Yes)
  - → P(Hot | No
  - → P(Mild | Yes)
  - → P(Cool | Yes)
  - → P(Mild | No)
  - → P(Mild | Yes)

- During training, it will calculate all these possible probabilities and store it in dictionary

EX → If new query pt occurs, example:

- ∴ {Sunny, Hot, mist, False} ⇒ X^T
- We have to predict whether it will be Yes or No
- 2 probabilities will be calculated as there is 2 classes



- $P(Yes | X^T)$ & $P(No | X_T)$
- $P(Yes | X_T) = P(X_1|Yes) \times P(X_2|Yes) \times P(X_3|Yes) \times P(X_4|Yes) \times P(Yes)$
- $P(Yes|X_T) = P(Sunny|Yes) \times P(Hot|Yes) \times P(mist|Yes) \times P(False|Yes) \times P(Yes)$
- And these all probabilities are

already stored in dictionary

- Similarly $P(No \mid x_T)$ will be calculated
- whoseever probab is highest, will be assigned that label
- refer to code example