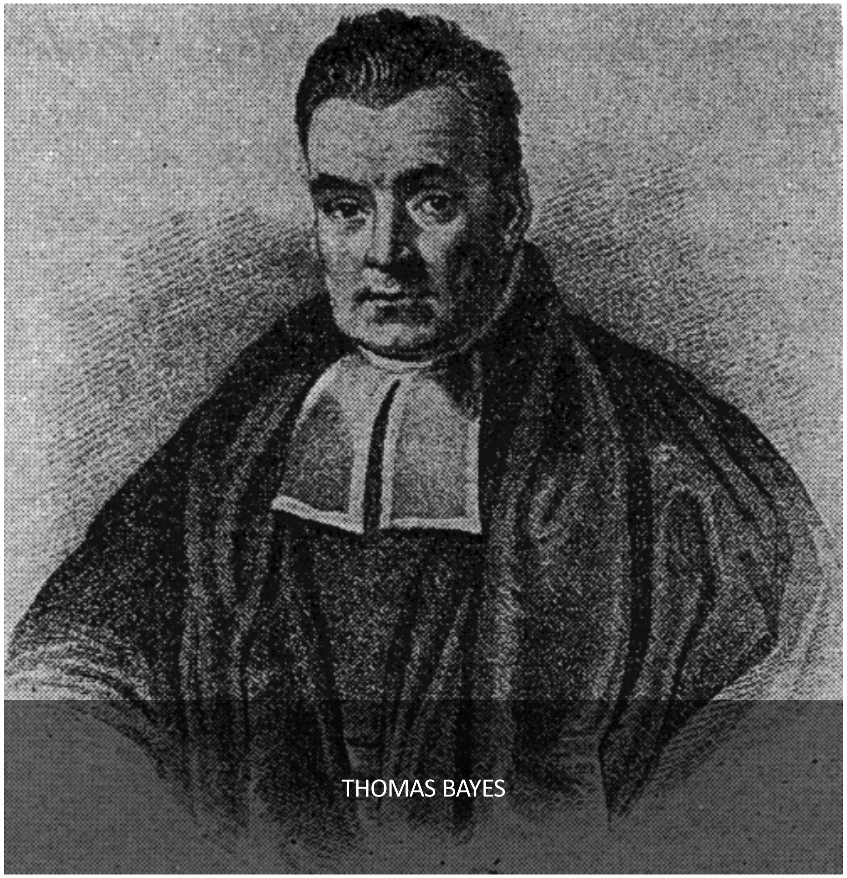
Machine Learning
CSCE 5215

Naïve Bayes Classifier

Instructor: Zeenat Tariq

Hello folks!
I am the one who formulated Bayes Theorem

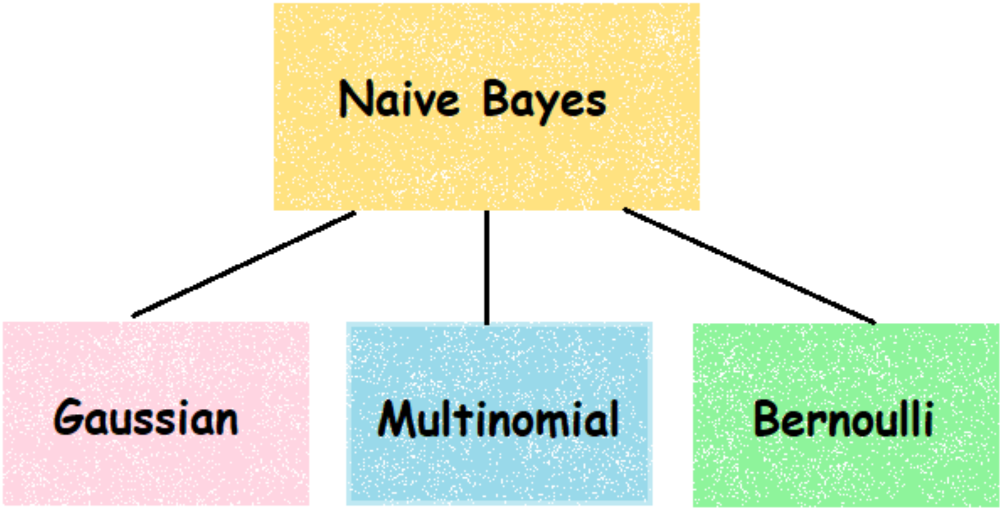


THOMAS BAYES

What is a Bayesian Model?

- Bayesian modeling is a statistical model where probability is influenced by the belief of the likelihood of a certain outcome.
- A Bayesian approach means that probabilities can be assigned to events that are neither repeatable nor random, such as the likelihood of a new novel becoming a *New York Times* bestseller.
- Naive Bayes classifier assumes features are independent of each other. Since that is rarely possible in real-life data, the classifier is called **naive**.

Types of Bayesian Model



Naïve Bayes

Asserts a global conditional independence between descriptive features given the target / class value

Step 1: Separate the training data by class

square root of the sum of the squared differences between the two vectors

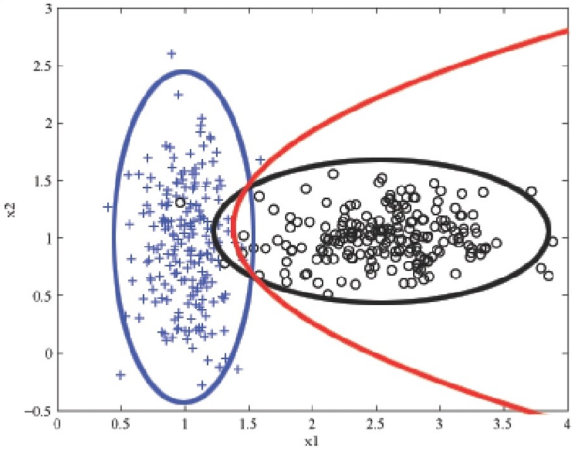
Step 2: Summarize the dataset

Calculate the mean and standard deviation of each input attribute / feature / column in the dataset

Step 3: Summarize the data by class

Step 4: Calculate the Gaussian Probability Density Function

Step 5: Calculate the class probabilities



Visually, Naive Bayes fits multidimensional gaussians to clouds of points to define a class.

Conditional Probability

Conditional probability is calculated by multiplying the probability of the preceding event by the updated probability of the succeeding, or conditional, event.

- **Event A** is that an individual applying for college will be **accepted**. There is an 80% chance that this individual will be accepted to college.

- **Event B** is that this individual will be given **dormitory housing**. Dormitory housing will only be provided for 60% of all of the accepted students.

- P (Accepted and dormitory housing) = P (Dormitory Housing | Accepted) P (Accepted) = (0.60)*(0.80) = 0.48.

A conditional probability would look at these two events in relationship with one another, such as the probability that you are both accepted to college, and you are provided with dormitory housing.

Bayes' Rule Applied to Documents and Classes

- Simple ("naive") classification method based on Bayes rule
- For a document *x* and a class *y*

$$P(Y|X) = \frac{P(X|Y)P(Y)}{P(X)}$$

Naive Bayes Classifier (I)

$$c_{MAP} = \underset{y \in C}{\operatorname{argmax}} P(y \mid x)$$

MAP is "maximum a posteriori" = most likely class

$$= \underset{y \in C}{\operatorname{argmax}} \frac{P(x \mid y)P(y)}{P(x)}$$

Bayes Rule

$$= \underset{y \in C}{\operatorname{argmax}} P(x \mid y)P(y)$$

Dropping the denominator

Naive Bayes Classifier (II)

"Likelihood" "Prior"

$$c_{MAP} = \underset{y \in C}{\operatorname{argmax}} \, P(x \mid y) P(y)$$

$$= \underset{y \in C}{\operatorname{argmax}} \, P(x_1, x_2, \ldots, x_n \mid y) P(y)$$

Document x represented as features x1..xn

Multinomial Naive Bayes

- The Multinomial Naive Bayes algorithm is a Bayesian learning approach popular in Natural Language Processing (NLP).
- The program guesses the tag of a text, such as an email or a newspaper story, using the Bayes theorem.
- It calculates each tag's likelihood for a given sample and outputs the tag with the greatest chance.

Multinomial Naive Bayes Classifier

$$c_{MAP} = \underset{y \in C}{\operatorname{argmax}} P(x_1, x_2, \ldots, x_n \mid y) P(y)$$

$$c_{NB} = \underset{y \in C}{\operatorname{argmax}} P(y_j) \prod_{x \in X} P(x \mid y)$$

Problems with multiplying lots of probs

- There's a problem with this:

$$c_{NB} = \underset{y_j \in C}{\operatorname{argmax}} P(y_j) \prod_{i \in positions} P(x_i \mid y_j)$$

- Multiplying lots of probabilities can result in floating-point underflow!
- .0006 * .0007 * .0009 * .01 * .5 * .000008….
- Idea: Use logs, because log(*ab*) = log(*a*) + log(*b*)
- We'll sum logs of probabilities instead of multiplying probabilities!

We actually do everything in log space

Instead of this: $$c_{NB} = \underset{c_j \in C}{\operatorname{argmax}} P(c_j) \prod_{i \in positions} P(x_i \mid c_j)$$

This: $$c_{\text{NB}} = \underset{c_j \in C}{\operatorname{argmax}} \left[\log P(c_j) + \sum_{i \in \text{positions}} \log P(x_i|c_j) \right]$$

Notes:

1) Taking log doesn't change the ranking of classes!

   The class with highest probability also has highest log probability!

2) It's a linear model:

   Just a max of a sum of weights: a **linear** function of the inputs

   So naive bayes is a **linear classifier**

Sec.13.3

Learning the Multinomial Naive Bayes Model

- First attempt: maximum likelihood estimates
  - simply use the frequencies in the data

$$\hat{P}(c_j) = \frac{N_{c_j}}{N_{total}}$$

$$\hat{P}(w_i \mid c_j) = \frac{count(w_i, c_j)}{\sum_{w \in V} count(w, c_j)}$$

Parameter estimation

$$\hat{P}(w_i \mid c_j) = \frac{count(w_i, c_j)}{\sum_{w \in V} count(w, c_j)}$$

fraction of times word w_i appears among all words in documents of topic c_j

- Create mega-document for topic *j* by concatenating all docs/data in this topic
  - Use frequency of *w* in mega-document

Problem with Maximum Likelihood

- What if we have seen no training documents with the word ***fantastic*** and classified in the topic **positive** (***thumbs-up)***?

$$\hat{P}(\text{"fantastic"} \mid \text{positive}) = \frac{count(\text{"fantastic", positive})}{\sum_{w \in V} count(w, \text{positive})} = 0$$

- Zero probabilities cannot be conditioned away, no matter the other evidence!

$$c_{MAP} = \text{argmax}_c \, \hat{P}(c) \prod_i \hat{P}(x_i \mid c)$$

What do I do when I get a Zero Probability???

Am here to solve your zero probability…

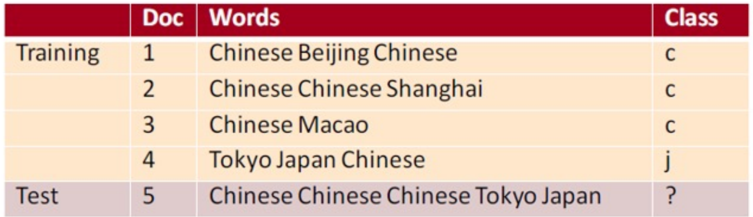
Laplace (add-1) smoothing for Naïve Bayes

$$\hat{P}(w_i \mid c) = \frac{count(w_i,c)+1}{\sum_{w \in V} \left(count(w,c) + 1 \right)}$$

$$= \frac{count(w_i,c)+1}{\left(\sum_{w \in V} count(w,c) \right) + |V|}$$

Let us look at a solved example

- We look at a Text Classification problem where we need to determine if a given sentence is Chinese or Japanese.



	Doc	Words	Class
Training	1	Chinese Beijing Chinese	c
	2	Chinese Chinese Shanghai	c
	3	Chinese Macao	c
	4	Tokyo Japan Chinese	j
Test	5	Chinese Chinese Chinese Tokyo Japan	?

- To classify the Test sentence, we need to find the probability of each word into each class. Here we have two classes, Chinese and Japanese.
- The class with highest probability value will be the determined class.
- Probability that the test sentence belongs to Chinese=

P(c) * P(Chinese/c)*P(Chinese/c)*P(Chinese/c)*P(Tokyo/c)*P(Japan/c)

P(Chinese/c)=5/8+6=5/14

P(Tokyo/c)= 0/8+6= 0 **oooopsss! We have encountered zero probability.**

$$\hat{P}(c) = \frac{N_c}{N}$$

$$\hat{P}(w \mid c) = \frac{count(w,c)+1}{count(c)+|V|}$$

	Doc	Words	Class
Training	1	Chinese Beijing Chinese	c
	2	Chinese Chinese Shanghai	c
	3	Chinese Macao	c
	4	Tokyo Japan Chinese	j
Test	5	Chinese Chinese Chinese Tokyo Japan	?

Priors:

$P(c)= \frac{3}{4}$

$P(j)= \frac{1}{4}$

Conditional Probabilities:

P(Chinese|*c*) = (5+1) / (8+6) = 6/14 = 3/7

P(Tokyo|*c*) = (0+1) / (8+6) = 1/14

P(Japan|*c*) = (0+1) / (8+6) = 1/14

P(Chinese|*j*) = (1+1) / (3+6) = 2/9

P(Tokyo|*j*) = (1+1) / (3+6) = 2/9

P(Japan|*j*) = (1+1) / (3+6) = 2/9

Choosing a class:

$P(c|d5) \propto 3/4 * (3/7)^3 * 1/14 * 1/14$

≈ 0.0003

$P(j|d5) \propto 1/4 * (2/9)^3 * 2/9 * 2/9$

≈ 0.0001

Multinomial Naïve Bayes: Learning

- From training corpus, extract *Vocabulary*

- Calculate $P(c_j)$ terms
  - For each c_j in C do

    $docs_j \leftarrow$ all docs with class $= c_j$

$$P(c_j) \leftarrow \frac{|docs_j|}{|\text{total \# documents}|}$$

- Calculate $P(w_k \mid c_j)$ terms
  - $Text_j \leftarrow$ single doc containing all $docs_j$
  - For each word w_k in *Vocabulary*

    $n_k \leftarrow$ # of occurrences of w_k in $Text_j$

$$P(w_k \mid c_j) \leftarrow \frac{n_k + \alpha}{n + \alpha |Vocabulary|}$$

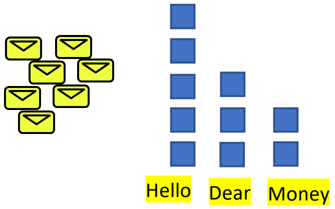
Let us learn the applications

Naive Bayes algorithm is used in the following places:

- **Face recognition**
- **Weather prediction**
- **Medical diagnosis**
- **Spam detection**
- **Age/gender identification**
- **Language identification**
- **Sentimental analysis**
- **Authorship identification**
- **News classification**

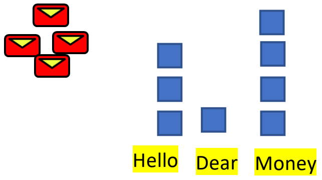
Application: Spam Filter

Histogram to calculate probability that you see a particular word GIVEN that the email you received is a NORMAL MESSAGE



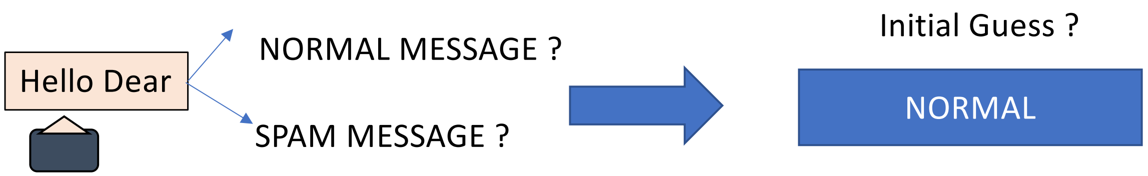
P(Hello | Normal) = 5/10 = 0.5
P(Dear | Normal) = 3/10 = 0.3
P(Money| Normal) = 2/10 = 0.2

Histogram to calculate probability that you see a particular word GIVEN that the email you received is a SPAM MESSAGE



P(Hello | Spam) = 3/8 = 0.37
P(Dear | Spam) = 1/8 = 0.12
P(Money| Spam) = 4/8 = 0.5

Application: Spam Filter



Hello Dear

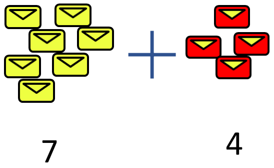
NORMAL MESSAGE ?

SPAM MESSAGE ?

Initial Guess ?

NORMAL

Training data
OR
Prior samples



P(Normal) = 7/11 = 0.63 →Prior probability
P(Spam) = 4/11 = 0.36

P(Hello | Normal) = 5/0 = 0.5
P(Dear | Normal) = 3/10 = 0.3
P(Money| Normal) = 2/10 = 0.2

Hello Dear

P(Hello | Spam) = 3/8 = 0.37
P(Dear | Spam) = 1/8=0.12
P(Money| Spam) = 4/8 = 0.5

P(Normal) x P(Hello | Normal) x P (Dear | Normal) = 0.63*0.5*0.2 = 0.063 ɑ P(Normal | Hello Dear)

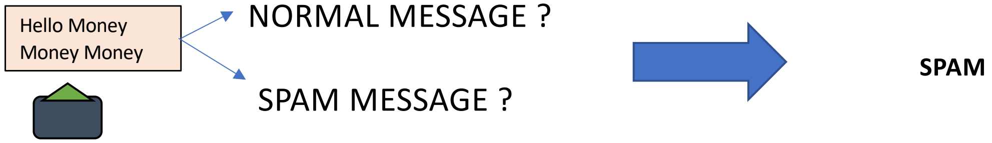
P(Spam) x P(Hello | Spam) x P (Dear | Spam) = 0.36*0.37*0.12 = 0.015 ɑ P(Spam | Hello Dear)

P(Normal | Hello Dear) ∝ P(Hello Dear | Normal) * P(Normal)

Generalized for classification: p(class | data) ∝ p(data | class) * p(class)

Pick the most probable class

Application: Spam Filter



Hello Money Money Money

NORMAL MESSAGE ?

SPAM MESSAGE ?

SPAM

Hello Money Money Money

Data

P(Normal) x P(Hello | Normal) x [P (Money | Normal)]^3= 0.63*0.5*0.2^3 = 0.0025

P(Spam) x P(Hello | Spam) x [P (Money | Spam)]^3 = 0.36*0.37*0.5^3 = **0.0166 > 0.0025**

Pick the largest probability to classify new data

Bernoulli Naïve Bayes

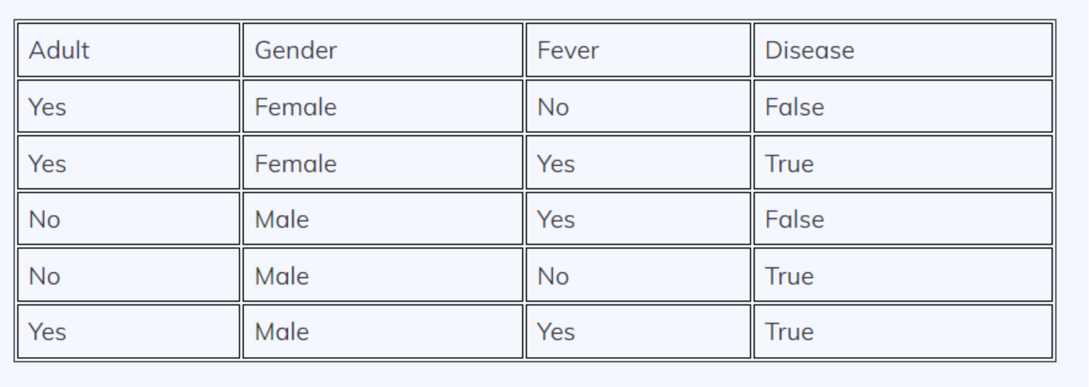
- Bernoulli Naive Bayes is a part of the Naive Bayes family. It is based on the Bernoulli Distribution and accepts only binary values, i.e., 0 or 1. If the features of the dataset are binary, then we can assume that Bernoulli Naive Bayes is the algorithm to be used.

- Example:

- (i) Bernoulli Naive Bayes classifier can be used to detect whether a person has a disease or not based on the data given. This would be a binary classification problem so that Bernoulli Naive Bayes would work well in this case.

- (ii) Bernoulli Naive Bayes classifier can also be used in text classification to determine whether an SMS is ‘spam’ or ‘not spam.

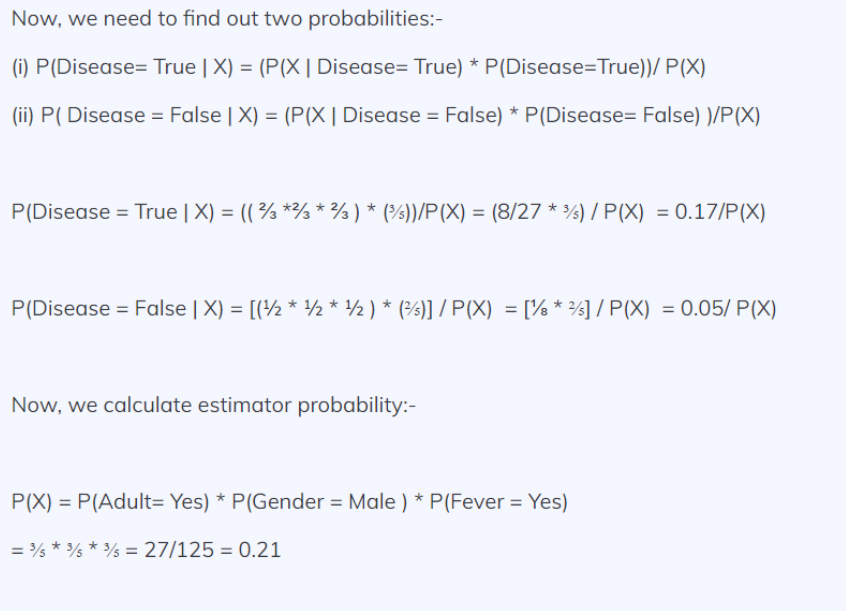
- Let us consider the example below to understand Bernoulli Naive Bayes:-



Adult	Gender	Fever	Disease
Yes	Female	No	False
Yes	Female	Yes	True
No	Male	Yes	False
No	Male	No	True
Yes	Male	Yes	True

In the above dataset, we are trying to predict whether a person has a disease or not based on their age, gender, and fever. Here, ‘Disease’ is the target, and the rest are the features.

- All values are binary.
- We wish to classify an instance 'X' where Adult='Yes', Gender= 'Male', and Fever='Yes'.
- Firstly, we calculate the class probability, probability of disease or not.



Now, we need to find out two probabilities:-

(i) P(Disease= True | X) = (P(X | Disease= True) * P(Disease=True))/ P(X)

(ii) P(Disease = False | X) = (P(X | Disease = False) * P(Disease= False))/P(X)

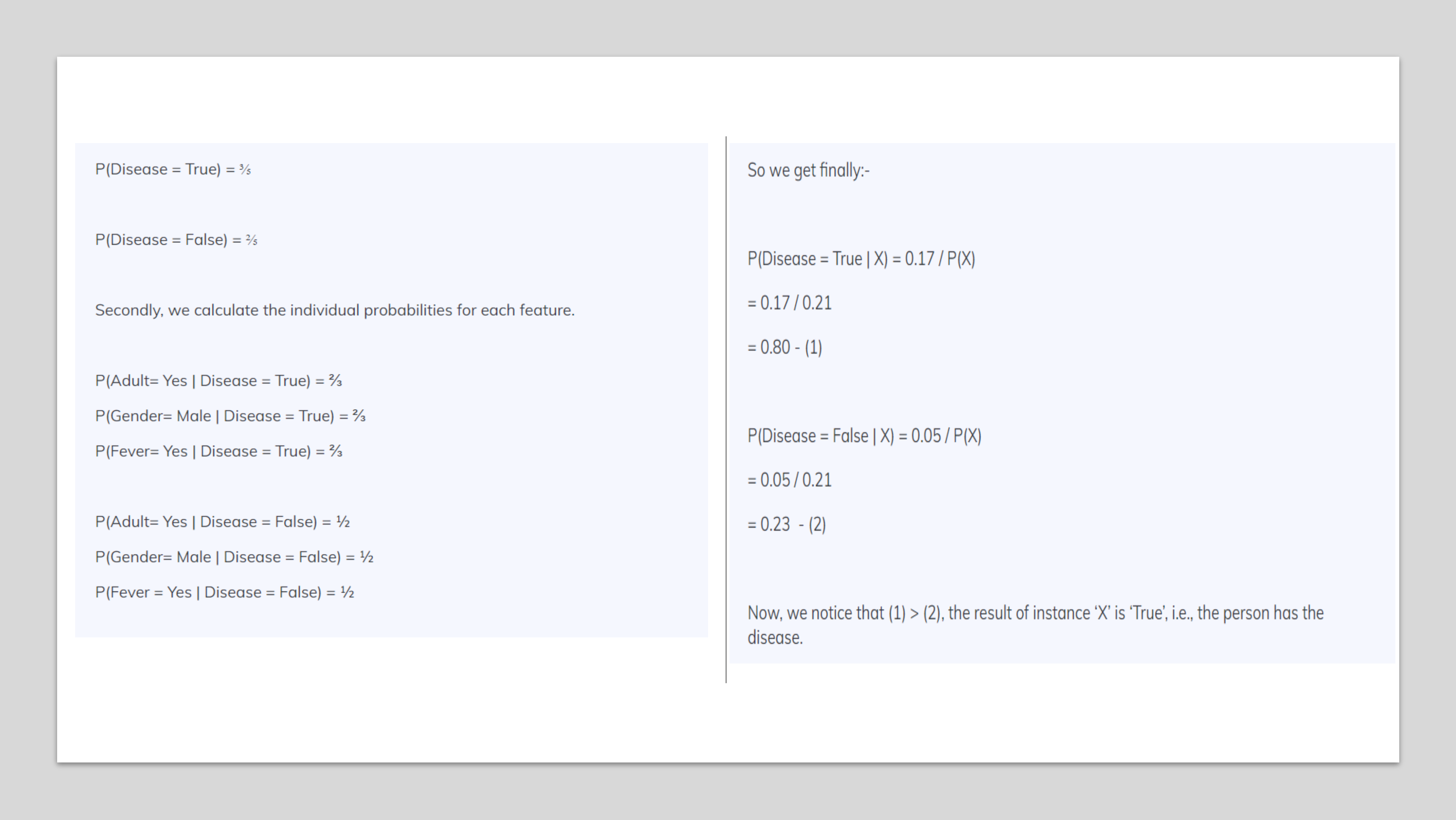
P(Disease = True | X) = ((⅔ *⅔ * ⅔) * (⅗))/P(X) = (8/27 * ⅗) / P(X) = 0.17/P(X)

P(Disease = False | X) = [(½ * ½ * ½) * (⅖)] / P(X) = [⅛ * ⅖] / P(X) = 0.05/ P(X)

Now, we calculate estimator probability:-

P(X) = P(Adult= Yes) * P(Gender = Male) * P(Fever = Yes)

= ⅗ * ⅗ * ⅗ = 27/125 = 0.21



P(Disease = True) = ⅗

P(Disease = False) = ⅖

Secondly, we calculate the individual probabilities for each feature.

P(Adult= Yes | Disease = True) = ⅔

P(Gender= Male | Disease = True) = ⅔

P(Fever= Yes | Disease = True) = ⅔

P(Adult= Yes | Disease = False) = ½

P(Gender= Male | Disease = False) = ½

P(Fever = Yes | Disease = False) = ½

So we get finally:-

P(Disease = True | X) = 0.17 / P(X)

= 0.17 / 0.21

= 0.80 - (1)

P(Disease = False | X) = 0.05 / P(X)

= 0.05 / 0.21

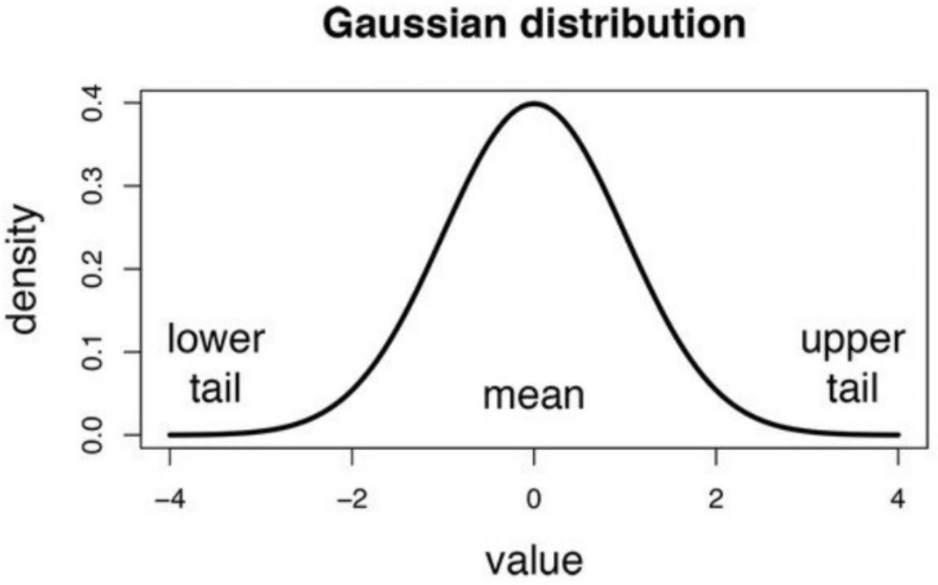
= 0.23 - (2)

Now, we notice that (1) > (2), the result of instance 'X' is 'True', i.e., the person has the disease.

Gaussian Naïve Bayes

- Gaussian Naïve Bayes is used when we assume all the continuous variables associated with each feature to be distributed according to **Gaussian Distribution.** Gaussian Distribution is also called Normal distribution.
- The conditional probability changes here since we have different values now. Also, the (PDF) probability density function of a normal distribution is given by:

$$P(x_i|y) = \frac{1}{\sqrt{2\pi\sigma_y^2}} exp\left(-\frac{(x_i - \mu_y)^2}{2\sigma_y^2}\right)$$

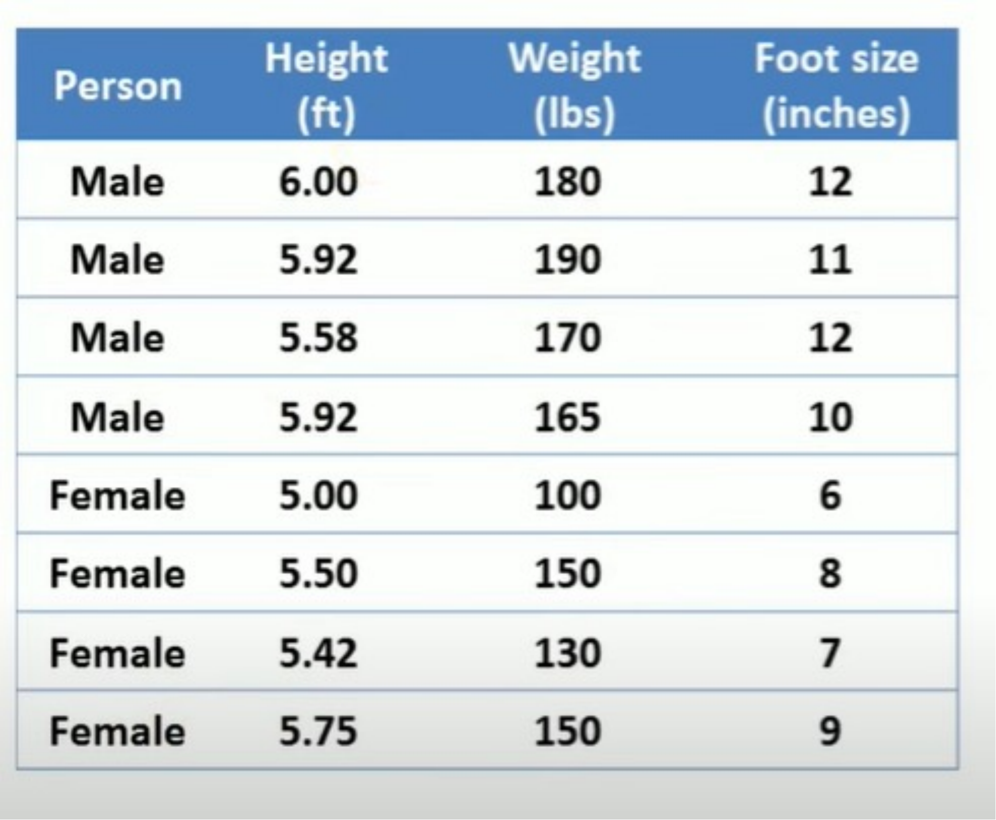


If we assume that events follow a Gaussian or normal distribution, we must use its probability density and call it Gaussian Naive Bayes.

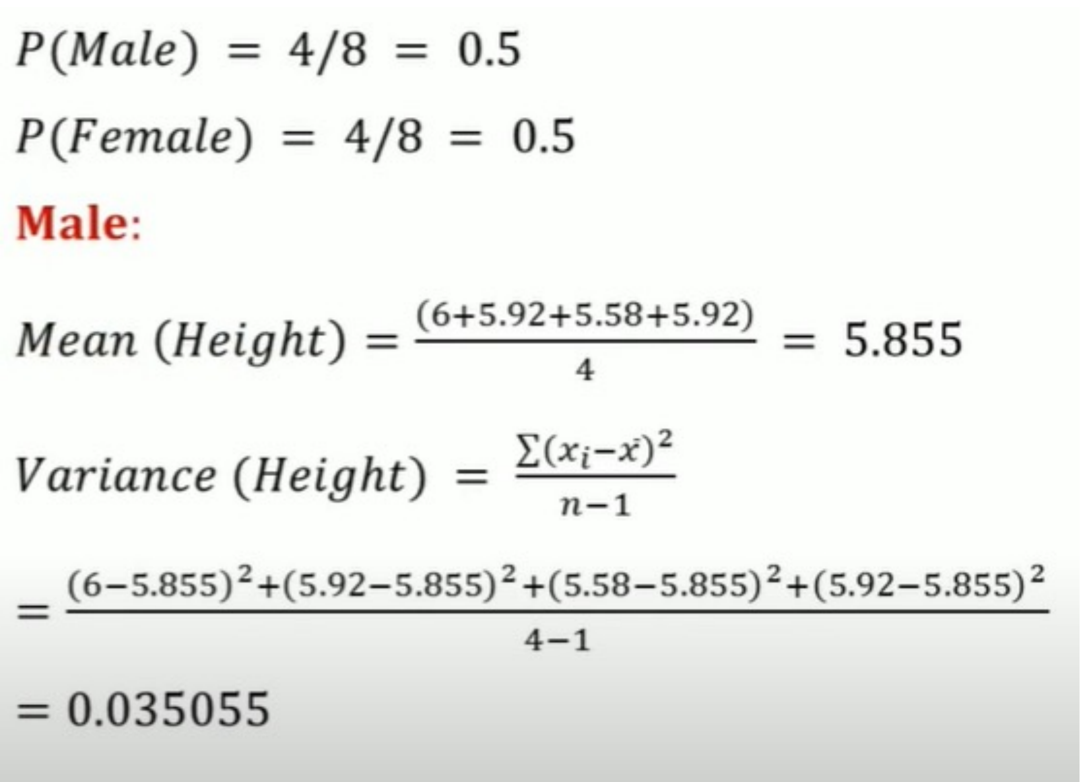
For the Naive gaussian Bayes, we use the following form:

$$P(X|Y=c)=\frac{1}{\sqrt{2\pi\sigma_c^2}}e^{\frac{-(x-\mu_c)^2}{2\sigma_c^2}}$$

- Based on the following data, determine gender of a person having 6ft height and 130lbs weight and 8 inch foot size.



Person	Height (ft)	Weight (lbs)	Foot size (inches)
Male	6.00	180	12
Male	5.92	190	11
Male	5.58	170	12
Male	5.92	165	10
Female	5.00	100	6
Female	5.50	150	8
Female	5.42	130	7
Female	5.75	150	9



$$P(Male) \;=\; 4/8 \;=\; 0.5$$

$$P(Female) \;=\; 4/8 \;=\; 0.5$$

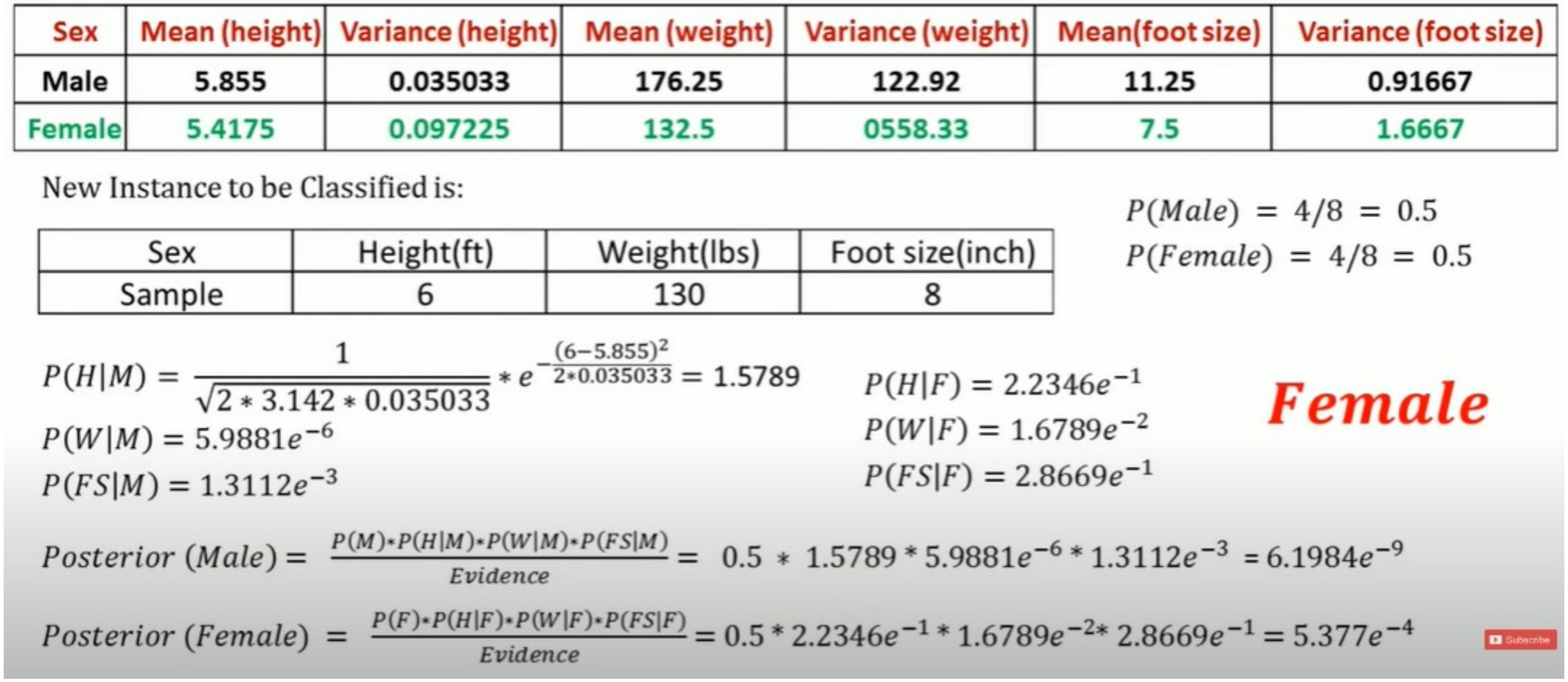
Male:

$$Mean\ (Height) = \frac{(6+5.92+5.58+5.92)}{4} \;=\; 5.855$$

$$Variance\ (Height) \;=\; \frac{\sum(x_i - \bar{x})^2}{n-1}$$

$$= \frac{(6-5.855)^2+(5.92-5.855)^2+(5.58-5.855)^2+(5.92-5.855)^2}{4-1}$$

$$= 0.035055$$



Sex	Mean (height)	Variance (height)	Mean (weight)	Variance (weight)	Mean(foot size)	Variance (foot size)
Male	5.855	0.035033	176.25	122.92	11.25	0.91667
Female	5.4175	0.097225	132.5	0558.33	7.5	1.6667

New Instance to be Classified is:

Sex	Height(ft)	Weight(lbs)	Foot size(inch)
Sample	6	130	8

$P(Male) = 4/8 = 0.5$

$P(Female) = 4/8 = 0.5$

$$P(H|M) = \frac{1}{\sqrt{2*3.142*0.035033}} * e^{-\frac{(6-5.855)^2}{2*0.035033}} = 1.5789$$

$P(W|M) = 5.9881e^{-6}$

$P(FS|M) = 1.3112e^{-3}$

$P(H|F) = 2.2346e^{-1}$

$P(W|F) = 1.6789e^{-2}$

$P(FS|F) = 2.8669e^{-1}$

Female

$$Posterior\ (Male) = \frac{P(M)*P(H|M)*P(W|M)*P(FS|M)}{Evidence} = 0.5 * 1.5789 * 5.9881e^{-6} * 1.3112e^{-3} = 6.1984e^{-9}$$

$$Posterior\ (Female) = \frac{P(F)*P(H|F)*P(W|F)*P(FS|F)}{Evidence} = 0.5 * 2.2346e^{-1} * 1.6789e^{-2} * 2.8669e^{-1} = 5.377e^{-4}$$

When to use?

Bernoulli Naive bayes is good at handling boolean/binary attributes, while **Multinomial Naive bayes** is good at handling discrete values and **Gaussian naive bayes** is good at handling continuous values.

Consider three scenarios:

1.Consider a dataset which has columns like has_diabetes, has_bp, has_thyroid and then you classify the person as healthy or not. In such a scenario **Bernoulli NB** will work well.

2.Consider a dataset that has marks of various students of various subjects and you want to predict, whether the student is clever or not. Then in this case **multinomial NB will** work fine.

3.Consider a dataset that has weight of students and you are predicting height of them, then **GaussiaNB** will well in this case.

Pros and Cons of NB

Advantages

- exceptionally fast training
- relative to other approaches, it works well when there is little data

Disadvantages

- independence assumption
- gaussians may not be appropriate to model the data distribution