Naive Bayes Classifier

INNOMATICS RESEARCH LABS

Overview

Naive Bayes is a supervised classification algorithm which belongs to the family of "**Probabilistic Classifiers**".

As the name suggests, it is uses Bayes' theorem at its core with a 'naive' assumption.

Naive assumption of conditional independence. The assumption is that all the features x1,x2...,xn are independent. This means that $P(A,B)=P(A).P(B)$.

Crux of the algorithm which is **Bayes rule** and it’s intuitive sense:

"You know something about the world, and based on what you know, you setup a probability model and you write down probabilities about the different outcomes. Then someone gives you some new information, which changes your beliefs and thus changes the probabilities of your outcomes."

Prof John Tsitsiklis (MITOpenCourseware).

Independent Events:

If the occurrence of one event doesn’t affect the occurrence of another event, then those events are considered as independent.

Ex: A= observing 6 on dice1, B = observing 5 on dice2, both are independent since the dice1 outcome doesn’t affect dice2.

$P(A \cap B) = P(A) * P(B)$

Similar to above instance, what if in case, we have only one dice

A=P(rolling 6 in dice1) and B=P(rolling 5 in dice1) then,
$P(A \cap B) = 0$

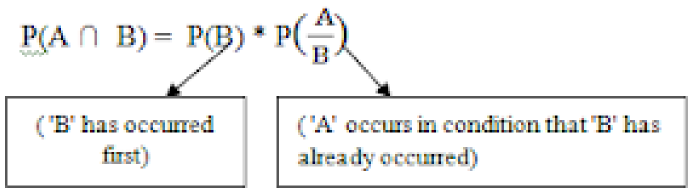
As both never occur at same time and same dice is being used for both events, these events are called as “**Mutually Exclusive events**”.

Conditional Probability: This is a measure of “probability of occurrence of one event assuming that another event has already occurred”. The probability of occurrence of A, when B has already occurred is denoted as

$$P\left(\frac{A}{B}\right) = \frac{P(A \cap B)}{P(B)} = \frac{P(\text{A and B jointly})}{P(B)}$$

conditional probability of A and B

In the above notation is explained as follows. From the simple definition of independence, if A, B are any two independent events, then
P(A ∩ B) = P(A) * P(B) (i.e. *Joint Probability*)
But in “*Joint probability”* we don’t know, whether these events are dependent or independent. So, the independence of these 2 events is:



$$P(A \cap B) = P(B) * P\left(\frac{A}{B}\right)$$

('B' has occurred first)

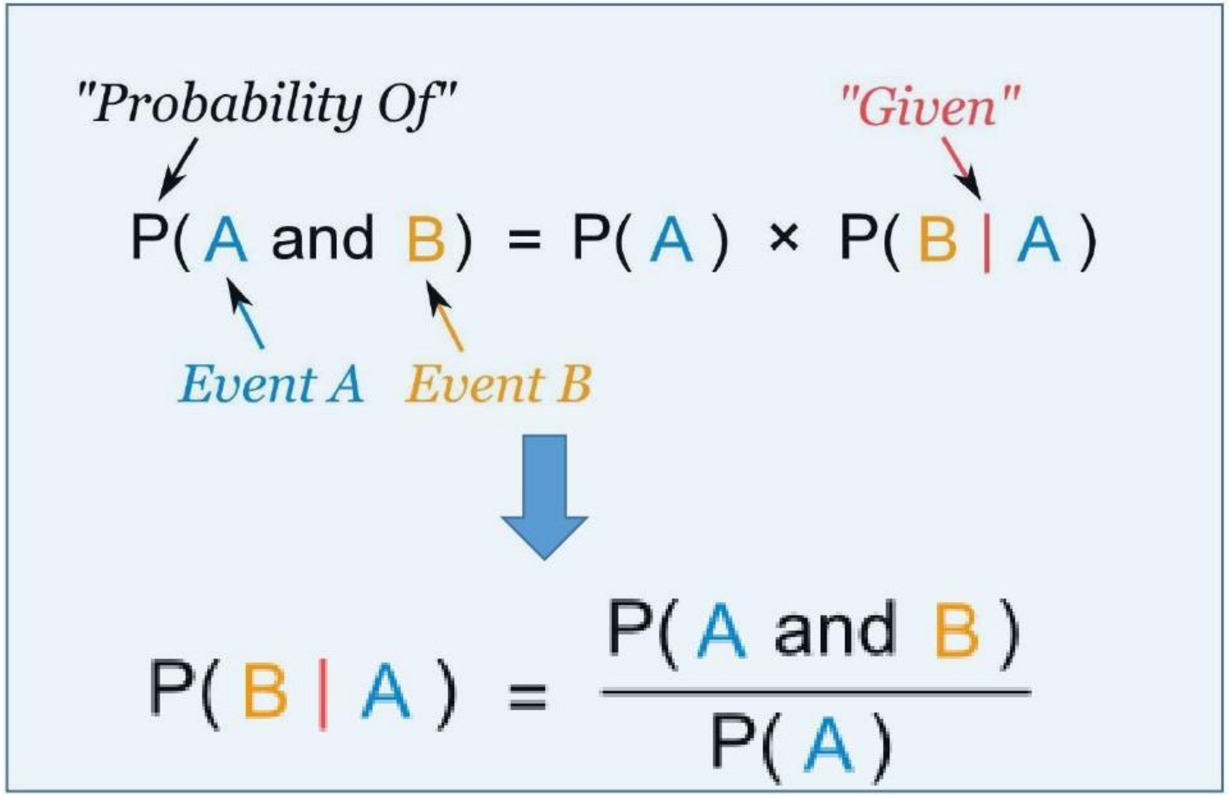
('A' occurs in condition that 'B' has already occurred)

Joint prob. of events A,B

Conditional Probability

P(A) means "Probability Of Event A"

Event A – Draw a Blue marble
Event B – Draw another Blue marble



"Probability Of"

"Given"

$$P(A \text{ and } B) = P(A) \times P(B \mid A)$$

Event A

Event B

$$P(B \mid A) = \frac{P(A \text{ and } B)}{P(A)}$$

P(A) = 2/5

P(B|A) = 1/4 ** **Blue Marble first**

P(B|A) = 2/4 ** **Red Marble first**

P(B|A) means "Event B **given** Event A"

Bayes Theorem

- Bayes’ Theorem is an extension of conditional probability.
- It allows us to find P(A|B) from P(B|A).

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

- P(A) is the prior probability or marginal probability of A. It is "prior" in the sense that it does not take into account any information about B.
- P(A|B) is the conditional probability of A, given B. It is also called the posterior probability because it is derived from or depends upon the specified value of B.
- P(B|A) is the conditional probability of B given A. It is also called the likelihood.
- P(B) is the prior or marginal probability of B, and acts as a normalizing constant.

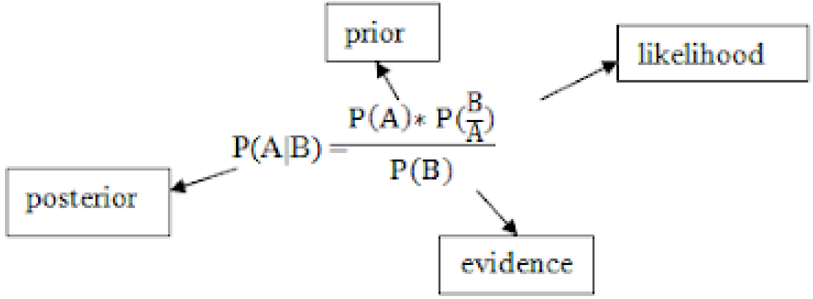
INNOMATICS RESEARCH LABS

Bayes Theorem: It is used to calculate the probability of an event based on its association with another event. It assumes all events are conditionally independent.

$$P\left(\frac{A}{B}\right) = \frac{P(A \cap B)}{P(B)} = \frac{P(B \cap A)}{P(B)}$$

Probability of A given B

After substituting “*joint probability p(A∩B)*” this in the above equation,



Probability of A given B

Conditional Independence:

A and B are said to be conditionally independent given C, if and only if

$$P\left(\frac{A \cap B}{C}\right)=P\left(\frac{A}{C}\right).\,P\left(\frac{B}{C}\right)$$

Prob. of (A∩B) given C

Where A, B are assumed to be independent.
Ex: A= Person *p1* probability of going late to home,
B= Person *p2* probability of going late to home,
C= There is a storm in the city.

First, solve the conditional probability of independent events A, B.

$$P\left(\frac{A}{B}\right) = \frac{P(A \cap B)}{P(B)} = \frac{P(A)*P(B)}{P(B)}$$

$$= P(A)$$

But, assume ‘C’ has given, So *P(Both goes late to home|it’s stormy) is*

$$P\left(\frac{A \cap B}{C}\right) = \frac{P(A \cap B \cap C)}{P(C)}$$

$$= \frac{P(C)*P\left(\frac{B}{C}\right)*P\left(\frac{A}{C}\right)}{P(C)} \text{(from conditional prob. above)}$$

$$=P\left(\frac{A}{C}\right).P\left(\frac{B}{C}\right)$$

Therefore, A, B are independent, given a condition ‘C’ (i.e, **conditionally independent**).

Suppose if event C= “*There is a storm in city and p1, p2 lives in same locality & uses same transport*”
then P(A|C) and P(B|C) are no more conditionally independent, because A,B are dependent now, since they’re from same locality.

Conditional independence on multiple events:

This derivation is going to be crucial, while we solve derivation of “*Naive Bayes algorithm*”.

Now, Let’s see how to extend the “*conditional independence for multiple events”* A,B,C,D,E,F……& on given some condition.

Assume some independent events A,B,C,D,E,F and given on a condition that event ‘Z’ has already occurred, as per conditional independence definition, all these becomes conditionally independent.

$$P\left(\frac{A\cap B\cap C\cap D\cap E\cap F}{Z}\right) = P\left(\frac{A}{Z}\right).P\left(\frac{B}{Z}\right).P\left(\frac{C}{Z}\right).P\left(\frac{D}{Z}\right)P\left(\frac{E}{Z}\right).P\left(\frac{F}{Z}\right)$$

conditional probability of events A,B....F given event Z

Finally! using these concepts, we reach the derivation of *“**Naive Bayes algorithm**”.*

In Machine learning “***Naive Bayes classifiers”*** are a family of simple probabilistic classifiers based on applying Bayes theorem with strong (naive) independence assumptions between the features.

All Naive Bayes classifiers assume that the value of a one feature is independent of the value of any other feature, on given a condition(i.e. class label).

For instance, a fruit may be considered to be an apple if it is red, round, and about 10 cm in diameter.

A naive Bayes classifier considers each of these features to contribute independently to the probability that this fruit is an apple, regardless of any possible correlations between the color, roundness, and diameter features.

Since, it assumes independence of events, Naive Bayes model serves as a benchmark model in most of the cases.

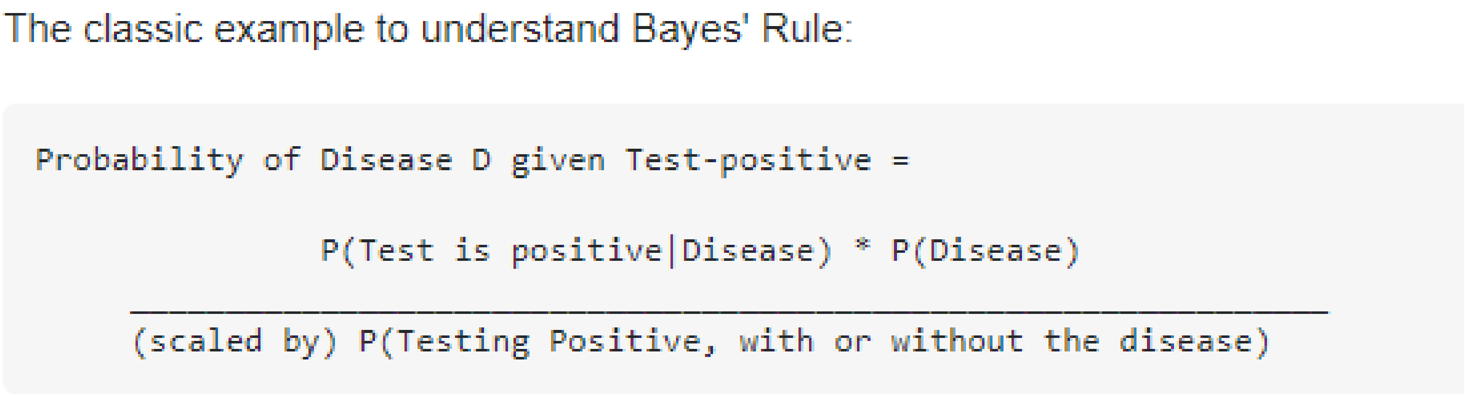
Understanding Bayes Rule

Conceptually, this is a way to go from P(Evidence| Known Outcome) to P(Outcome|Known Evidence).

Often, we know how frequently some particular evidence is observed, *given a known outcome*.

We have to use this known fact to compute the reverse, to compute the chance of that *outcome happening*, given the evidence.

P(Outcome given that we know some Evidence) =
P(Evidence given that we know the Outcome) times Prob(Outcome), scaled by the P(Evidence)



The classic example to understand Bayes' Rule:

```
Probability of Disease D given Test-positive =

               P(Test is positive|Disease) * P(Disease)
     _______________________________________________________________
     (scaled by) P(Testing Positive, with or without the disease)
```

Naïve Bayes Classifier

- The Bayes Rule provides the formula for the probability of Y given X. But, in real-world problems, you typically have multiple X variables.
- When the features are independent, we can extend the Bayes Rule to what is called Naive Bayes.
- It is called 'Naive' because of the naïve (simple) assumption that the X's are independent of each other.

Naïve Bayes Classifier

Bayes theorm in the terms of dependent & independent variables -

$$P\ (Y = k \mid X) = \frac{P(X \mid Y = k)\ \times\ P(Y = k)}{P(X)}$$

Where, k is a class of Y.

Now, Naïve Bayes- equation is –

$$\boldsymbol{P\ (Y = k \mid X1, X2 \ldots.Xn)} = \frac{\boldsymbol{P(X1 \mid Y = k)} \times \boldsymbol{P(X2 \mid Y = k)} \times \cdots \times \boldsymbol{P(Xn \mid Y = k)} \times \boldsymbol{P(Y = k)}}{\boldsymbol{P(X1)} \times \boldsymbol{P(X2)} \ldots. \times \boldsymbol{P(Xn)}}$$

INNOMATICS
RESEARCH LABS

Naïve Bayes Classifier

Terminology:

- $\boldsymbol{P\ (Y = k \mid X1, X2 \ldots . Xn)}$ – Posterior probability or Posterior.
- $\boldsymbol{P(X1 \mid Y = k)\ \times P(X2 \mid Y = k) \times \cdots \times P(Xn \mid Y = k)}$ – (Probability) Likelihood of Evidence.
- $\boldsymbol{P(Y = k)}$ – Prior. Overall probability of Y=k. Simply, Prior = count(Y=k)/n.
- $\boldsymbol{P(X1) \times P(X2) \ldots . \times P(Xn)}$ = Probability of Evidence.

Naïve Bayes Classifier

posterior probability

Likelihood

Prior *probability*

$$P(Y|X_1,\ldots,X_n) = \frac{P(X_1,\ldots,X_n|Y)P(Y)}{P(X_1,\ldots,X_n)}$$

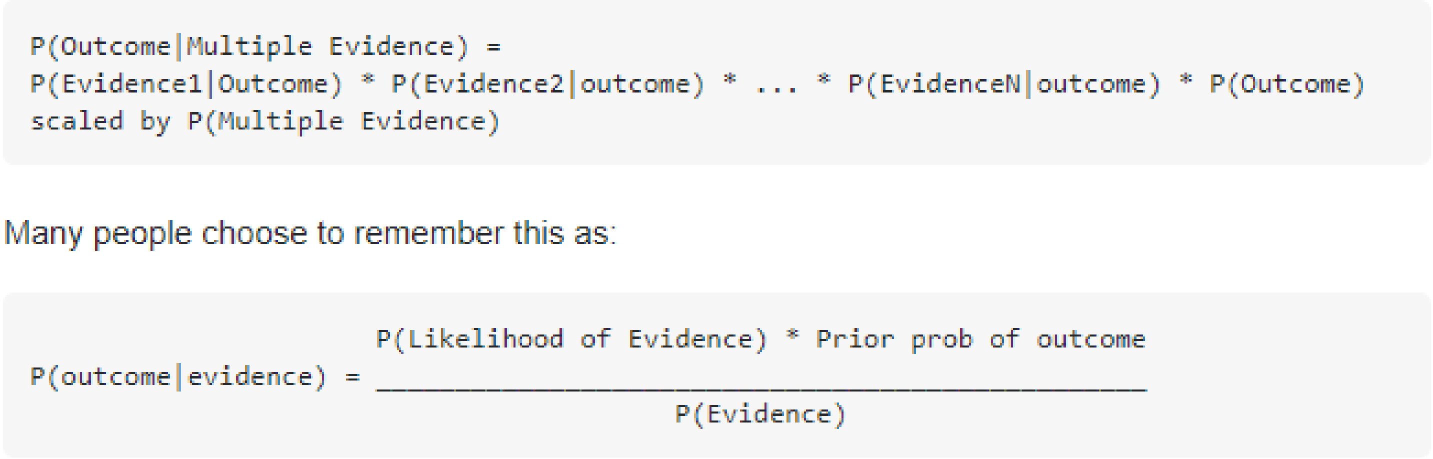
Evidence

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

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

Getting to Naive Bayes

So far, we have talked only about one piece of evidence. In reality, we have to predict an outcome given **multiple evidence.** In that case, the math gets very complicated. To get around that complication, one approach is to 'uncouple' multiple pieces of evidence, and to treat each of piece of evidence as independent. This approach is why this is called *naive* Bayes.



```
P(Outcome|Multiple Evidence) =
P(Evidence1|Outcome) * P(Evidence2|outcome) * ... * P(EvidenceN|outcome) * P(Outcome)
scaled by P(Multiple Evidence)
```

Many people choose to remember this as:

```
                      P(Likelihood of Evidence) * Prior prob of outcome
P(outcome|evidence) = _________________________________________________
                                         P(Evidence)
```

Notice a few things about this equation:

- If the Prob(evidence|outcome) is 1, then we are just multiplying by 1.
- If the Prob(some particular evidence|outcome) is 0, then the whole prob. becomes 0.
- If you see contradicting evidence, we can rule out that outcome.
- Since we divide everything by P(Evidence), we can even get away without calculating it.
- The intuition behind multiplying by the *prior* is so that we give high probability to more common outcomes, and low probabilities to unlikely outcomes. These are also called base rates and they are a way to scale our predicted probabilities.

How to Apply NaiveBayes to Predict an Outcome?

Just run the formula above for each possible outcome. Since we are trying to *classify*, each outcome is called a class and it has a class label.

Our job is to look at the evidence, to consider how likely it is to be this class or that class, and assign a label to each entity.

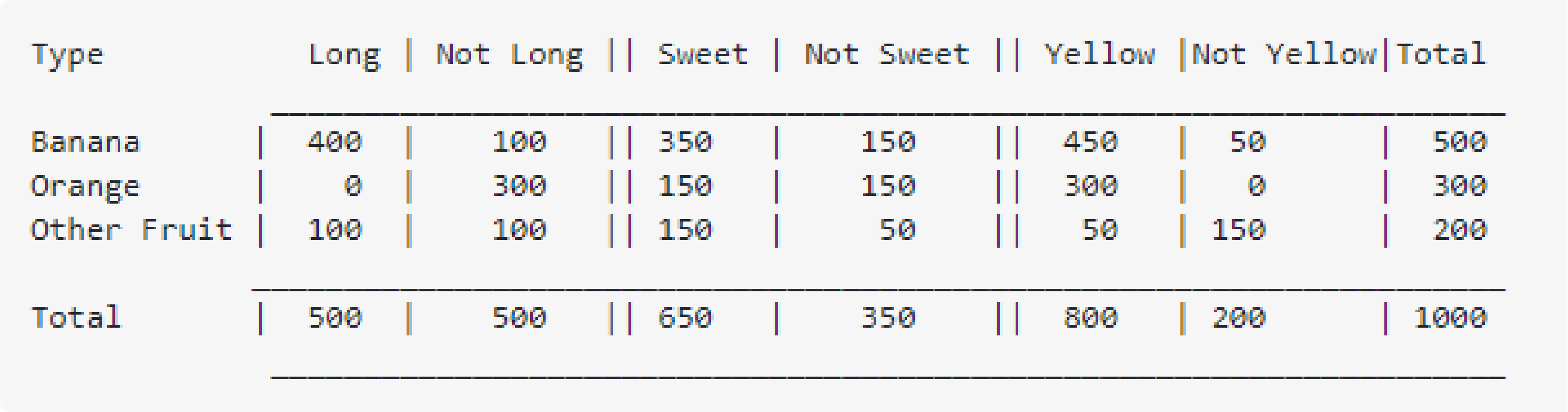
Again, we take a very simple approach: The class that has the highest probability is declared the "winner" and that class label gets assigned to that combination of evidences.

Fruit Example

Let's try it out on an example to increase our understanding: The OP asked for a 'fruit' identification example. Let's say that we have data on 1000 pieces of fruit. They happen to be **Banana**, **Orange** or some **Other Fruit**. We know 3 characteristics about each fruit:

1. Whether it is Long
2. Whether it is Sweet and
3. If its color is Yellow.

This is our 'training set.' We will use this to predict the type of any *new* fruit we encounter.

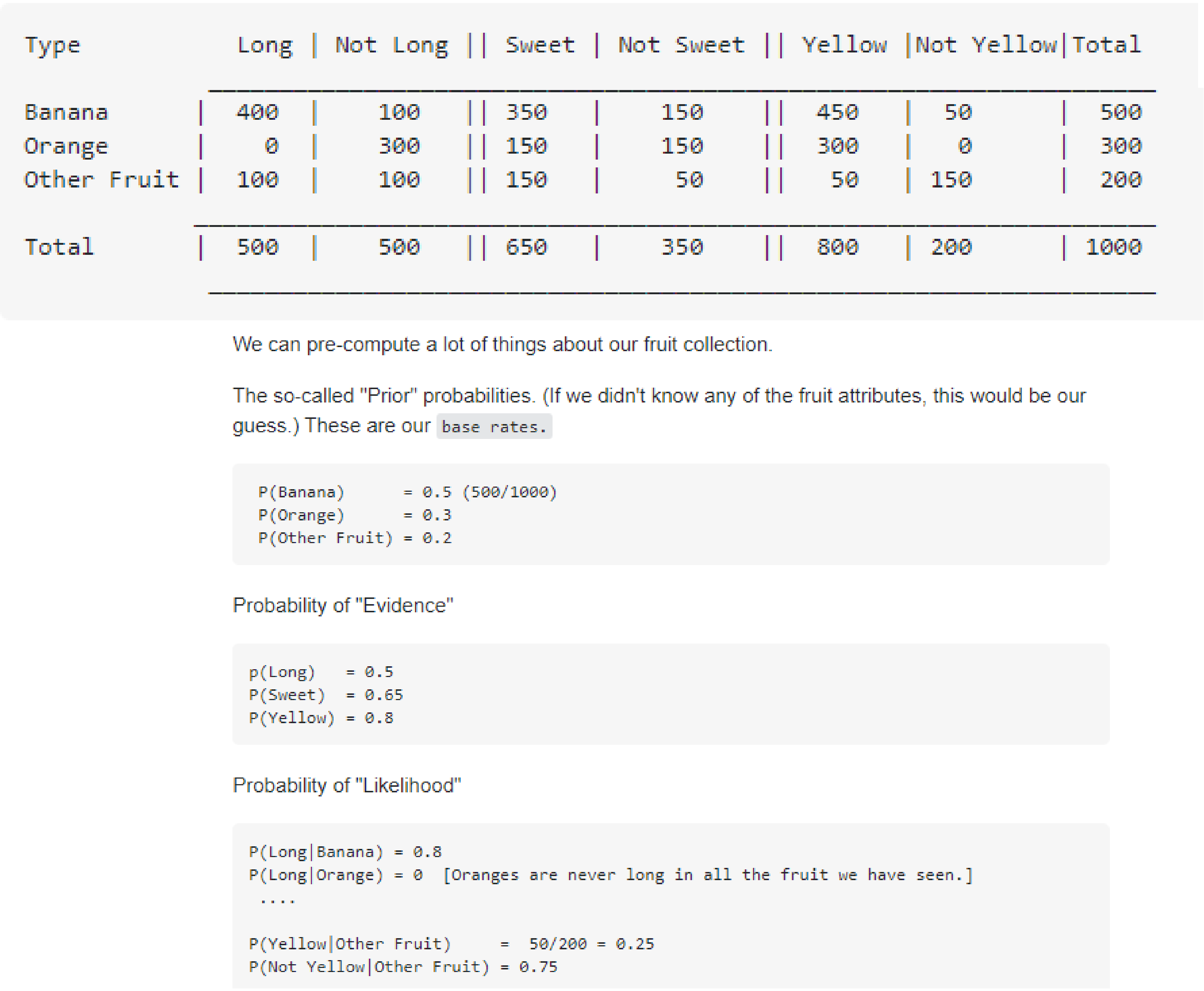


```
Type           Long | Not Long || Sweet | Not Sweet || Yellow |Not Yellow|Total
               ___________________________________________________________________

Banana      | 400  |     100  || 350   |     150   ||  450   |  50      |  500
Orange      |   0  |     300  || 150   |     150   ||  300   |   0      |  300
Other Fruit | 100  |     100  || 150   |      50   ||   50   | 150      |  200
             ____________________________________________________________________

Total       | 500  |     500  || 650   |     350   ||  800   | 200      | 1000
               ___________________________________________________________________
```

Reference: Article on medium.com



Type	Long	Not Long	Sweet	Not Sweet	Yellow	Not Yellow	Total
Banana	400	100	350	150	450	50	500
Orange	0	300	150	150	300	0	300
Other Fruit	100	100	150	50	50	150	200
Total	500	500	650	350	800	200	1000

We can pre-compute a lot of things about our fruit collection.

The so-called "Prior" probabilities. (If we didn't know any of the fruit attributes, this would be our guess.) These are our `base rates.`

```
P(Banana)      = 0.5 (500/1000)
P(Orange)      = 0.3
P(Other Fruit) = 0.2
```

Probability of "Evidence"

```
p(Long)   = 0.5
P(Sweet)  = 0.65
P(Yellow) = 0.8
```

Probability of "Likelihood"

```
P(Long|Banana) = 0.8
P(Long|Orange) = 0  [Oranges are never long in all the fruit we have seen.]
 ....

P(Yellow|Other Fruit)     =  50/200 = 0.25
P(Not Yellow|Other Fruit) = 0.75
```

Reference: Article on medium.com

Given a Fruit, how to classify it?

Let's say that we are given the properties of an unknown fruit, and asked to classify it.

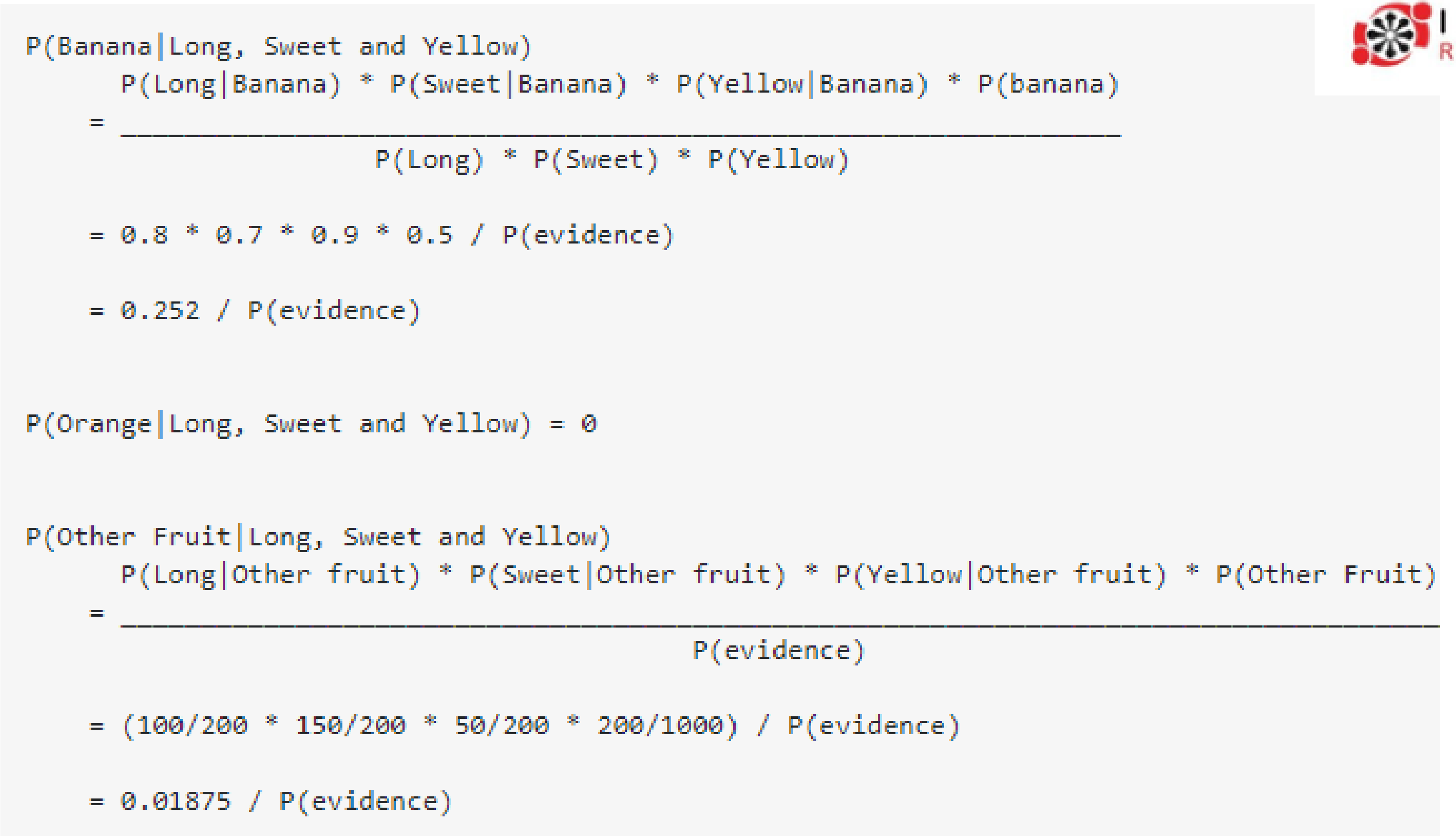
We are told that the fruit is Long, Sweet and Yellow.

Is it a Banana? Is it an Orange? Or Is it some Other Fruit?

We can simply run the numbers for each of the 3 outcomes, one by one.

Then we choose the highest probability and 'classify' our unknown fruit as belonging to the class that had the highest probability based on our prior evidence

(our 1000 fruit training set):



```
P(Banana|Long, Sweet and Yellow)
      P(Long|Banana) * P(Sweet|Banana) * P(Yellow|Banana) * P(banana)
    = _______________________________________________________________
                      P(Long) * P(Sweet) * P(Yellow)

    = 0.8 * 0.7 * 0.9 * 0.5 / P(evidence)

    = 0.252 / P(evidence)


P(Orange|Long, Sweet and Yellow) = 0


P(Other Fruit|Long, Sweet and Yellow)
      P(Long|Other fruit) * P(Sweet|Other fruit) * P(Yellow|Other fruit) * P(Other Fruit)
    = ____________________________________________________________________________________
                                          P(evidence)

    = (100/200 * 150/200 * 50/200 * 200/1000) / P(evidence)

    = 0.01875 / P(evidence)
```

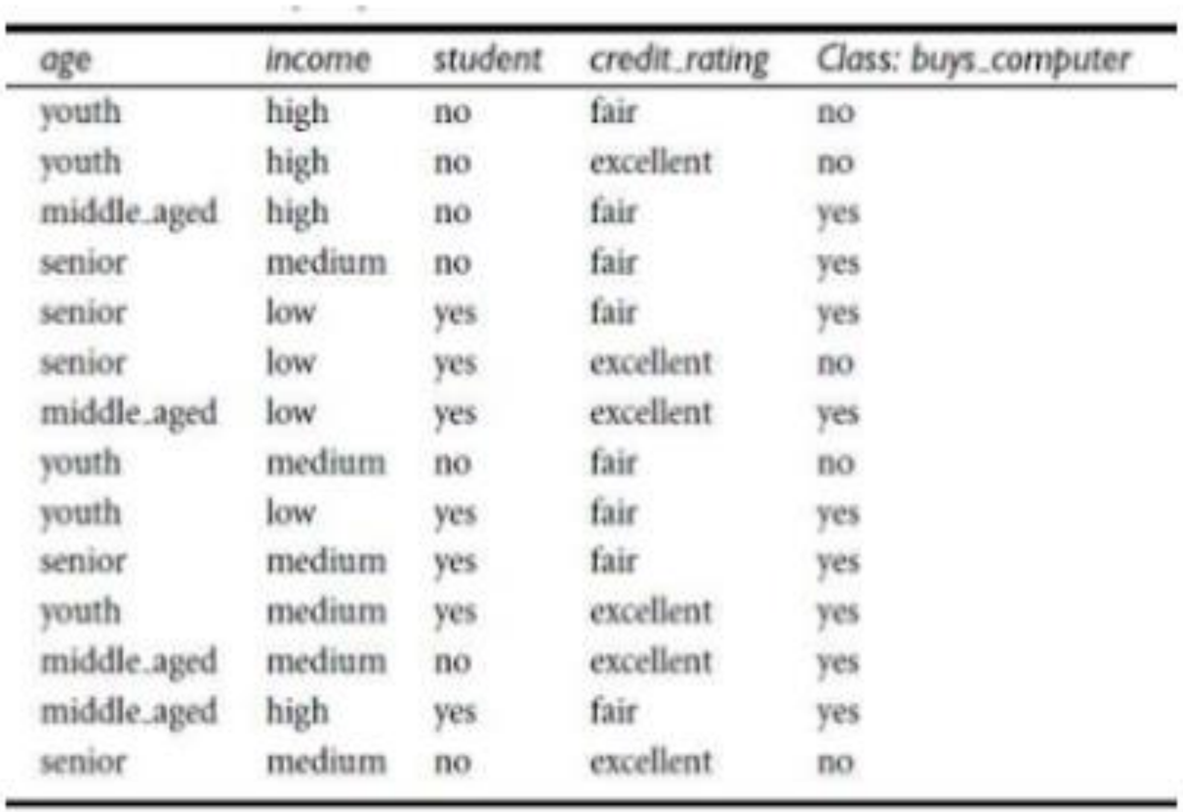
By an overwhelming margin (0.252 >> 0.01875), we classify this Sweet/Long/Yellow fruit as likely to be a **Banana**.

Reference: Article on medium.com

Computer Example:

The 4 features <f1, f2, f3, f4> are <‘age’, ‘income’, ‘yes’, ‘credit_rating’>.
If we have a query point from the test data Xq<youth, high, yes, excellent> , predict whether buys a computer or not?

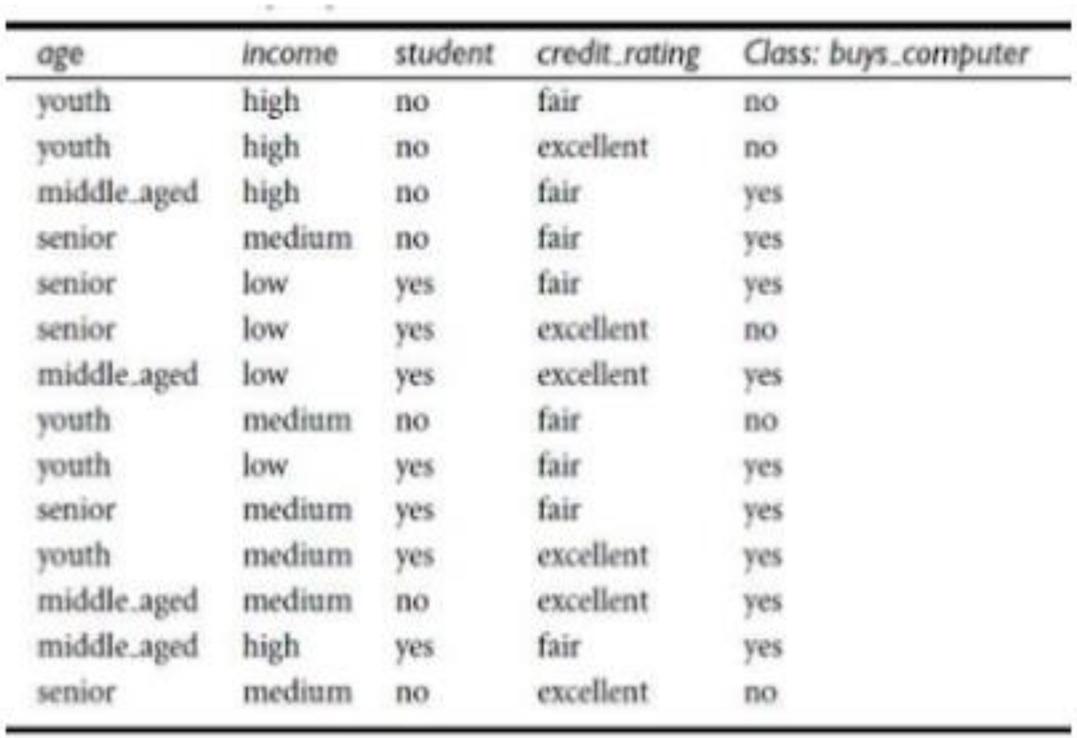
$$\text{Posterior for buys\_computer} = \max(\,P(\text{buys\_computer}_{yes}),\ P(\text{buys\_computer}_{no})\,)\ \text{given}\ X_q$$



age	*income*	*student*	*credit_rating*	*Class: buys_computer*
youth	high	no	fair	no
youth	high	no	excellent	no
middle_aged	high	no	fair	yes
senior	medium	no	fair	yes
senior	low	yes	fair	yes
senior	low	yes	excellent	no
middle_aged	low	yes	excellent	yes
youth	medium	no	fair	no
youth	low	yes	fair	yes
senior	medium	yes	fair	yes
youth	medium	yes	excellent	yes
middle_aged	medium	no	excellent	yes
middle_aged	high	yes	fair	yes
senior	medium	no	excellent	no

Reference: Article on medium.com

Step 1: *Calculate the posterior probability for buys_computer = yes , given Xq <youth, high, student, excellent>.*



age	income	student	credit_rating	Class: buys_computer
youth	high	no	fair	no
youth	high	no	excellent	no
middle_aged	high	no	fair	yes
senior	medium	no	fair	yes
senior	low	yes	fair	yes
senior	low	yes	excellent	no
middle_aged	low	yes	excellent	yes
youth	medium	no	fair	no
youth	low	yes	fair	yes
senior	medium	yes	fair	yes
youth	medium	yes	excellent	yes
middle_aged	medium	no	excellent	yes
middle_aged	high	yes	fair	yes
senior	medium	no	excellent	no

$$P\left(\frac{\text{buys\_computer}_{yes}}{\text{Youth}\cap\text{High}\cap\text{Student}\cap\text{Excellent}}\right) = P\left(\frac{youth}{\text{buys\_computer}_{yes}}\right) * P\left(\frac{high}{\text{buys\_computer}_{yes}}\right) *$$

$$P\left(\frac{student}{\text{buys\_computer}_{yes}}\right) * P\left(\frac{youth}{\text{buys\_computer}_{yes}}\right) * P(\text{buys\_computer}_{yes})$$

Prob. of buys_computer =yes, given all features

$$P\left(\frac{youth}{\text{buys\_computer}_{yes}}\right) = \frac{\text{count of age}_{\text{youth}}\&\text{buys\_computer}_{\text{yes}}}{\text{count of buys\_computer}_{\text{yes}}} = 2/9$$

$$P\left(\frac{high}{\text{buys\_computer}_{yes}}\right) = \frac{\text{count of income}_{\text{high}}\&\text{buys\_computer}_{\text{yes}}}{\text{count of buys\_computer}_{\text{yes}}} = 2/9$$

$$P\left(\frac{student}{\text{buys\_computer}_{yes}}\right) = \frac{\text{count of student}_{\text{yes}}\&\text{buys\_computer}_{\text{yes}}}{\text{count of buys\_computer}_{\text{yes}}} = 6/9$$

$$P\left(\frac{credit\_rating}{\text{buys\_computer}_{yes}}\right) = \frac{\text{count of credit\_rating}_{\text{excellent}}\&\text{buys\_computer}_{\text{yes}}}{\text{count of buys\_computer}_{\text{yes}}} = 3/9$$

$$P(\text{buys\_computer}_{yes}) = 9/14$$

Multiplying all these likely hoods,Posterior for $\text{buys\_computer}_{yes} = 0.0071$

Reference: Article on medium.com

***Step 2:** Similarly find the posterior for buys_computer = no.*

$$\text{Posterior for buys\_computer}_{\text{no}} = \text{P}\left(\frac{\text{buys\_computer}_{\text{no}}}{\text{Youth} \cap \text{High} \cap \text{Student} \cap \text{Excellent}} \right)$$

Multiplying all these likely hoods, the posterior of buys_computer=no is **0.0102**

Therefore, max(0.0071, 0.0102) shows, predicted class label=”no”,

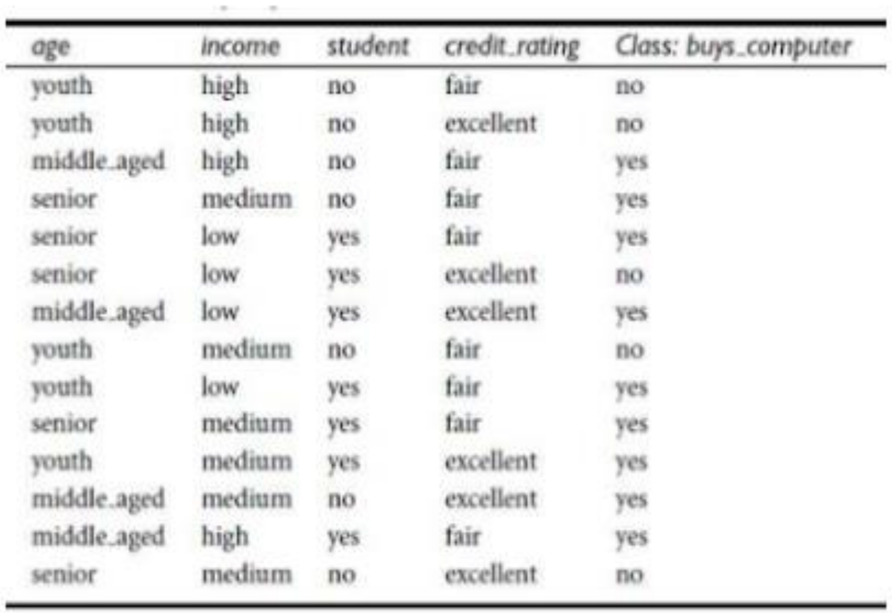
So that student doesn’t buy computer.

Reference: Article on medium.com

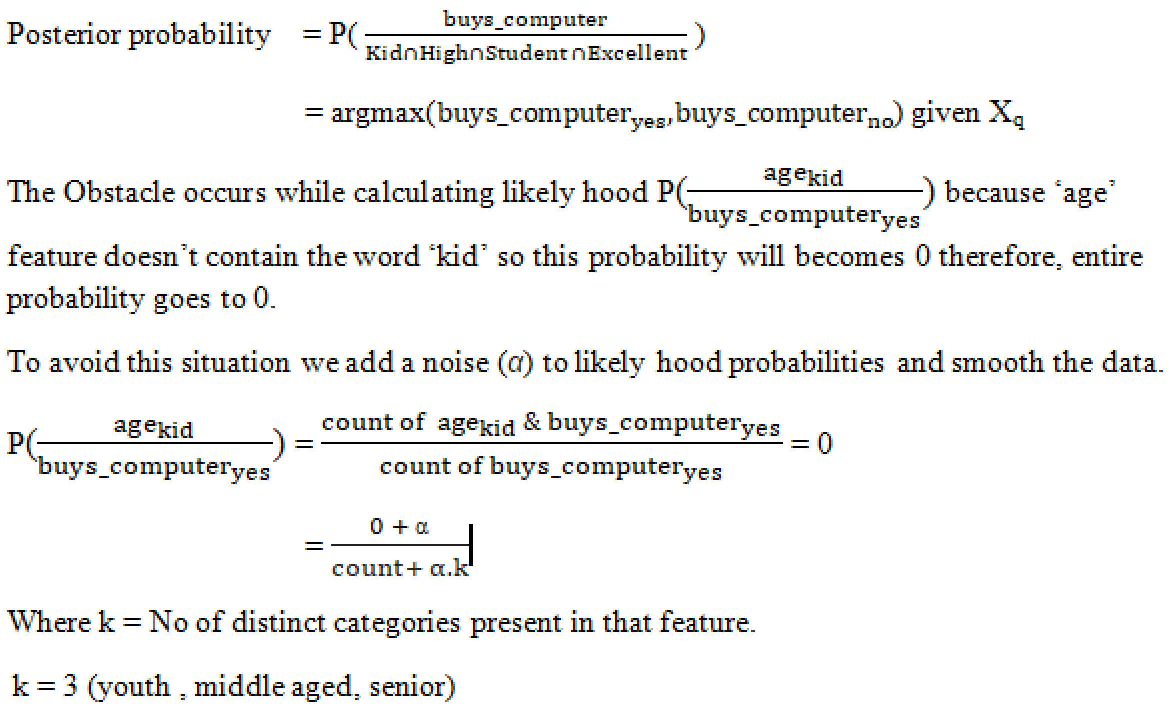
Laplace (or) Additive Smoothing:

This is introduced to solve the problem of **zero probability** - *“If query point contains a new observation, which is not yet seen in training data while calculating probabilities”*.

Let’s deal this with same dataset, but with a different query point Xq.
Xq<kid, high, yes, excellent>



age	income	student	credit_rating	Class: buys_computer
youth	high	no	fair	no
youth	high	no	excellent	no
middle_aged	high	no	fair	yes
senior	medium	no	fair	yes
senior	low	yes	fair	yes
senior	low	yes	excellent	no
middle_aged	low	yes	excellent	yes
youth	medium	no	fair	no
youth	low	yes	fair	yes
senior	medium	yes	fair	yes
youth	medium	yes	excellent	yes
middle_aged	medium	no	excellent	yes
middle_aged	high	yes	fair	yes
senior	medium	no	excellent	no



Posterior probability $= P(\frac{buys\_computer}{Kid \cap High \cap Student \cap Excellent})$

$= argmax(buys\_computer_{yes}, buys\_computer_{no})$ given X_q

The Obstacle occurs while calculating likely hood $P(\frac{age_{kid}}{buys\_computer_{yes}})$ because ‘age’ feature doesn’t contain the word ‘kid’ so this probability will becomes 0 therefore, entire probability goes to 0.

To avoid this situation we add a noise (α) to likely hood probabilities and smooth the data.

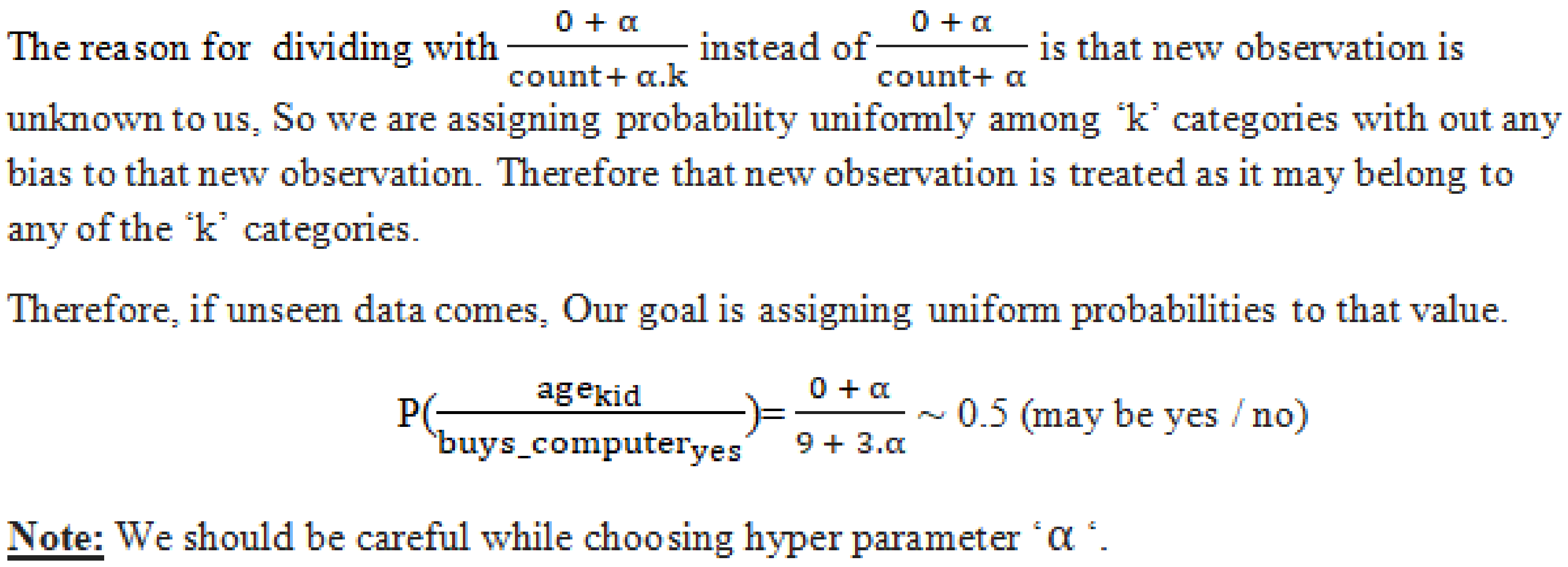
$$P(\frac{age_{kid}}{buys\_computer_{yes}}) = \frac{\text{count of } age_{kid} \,\&\, buys\_computer_{yes}}{\text{count of } buys\_computer_{yes}} = 0$$

$$= \frac{0+\alpha}{count + \alpha.k}$$

Where k = No of distinct categories present in that feature.

k = 3 (youth , middle aged, senior)

Reference: Article on medium.com



The reason for dividing with $\frac{0+\alpha}{\text{count}+\alpha.k}$ instead of $\frac{0+\alpha}{\text{count}+\alpha}$ is that new observation is unknown to us, So we are assigning probability uniformly among 'k' categories with out any bias to that new observation. Therefore that new observation is treated as it may belong to any of the 'k' categories.

Therefore, if unseen data comes, Our goal is assigning uniform probabilities to that value.

$$P(\frac{\text{age}_{\text{kid}}}{\text{buys\_computer}_{\text{yes}}}) = \frac{0+\alpha}{9+3.\alpha} \sim 0.5 \text{ (may be yes / no)}$$

Note: We should be careful while choosing hyper parameter 'α'.

alpha is the noise- hyper parameter

Precautions:
'α' should not be too high or too small, should be chosen properly by taking "*Bias-variance*" trade off into consideration.
'α' should not disturb the uniform probabilities that are assigned to unknown data/new observations.

Finding Optimal 'α':
Using elbow plot, try plotting 'performance metric' v/s 'α' hyper-parameter.

Reference: Article on medium.com

Applications of Naïve Bayes

- Spam Classification
  - Given an email, predict whether it is spam or not
- Medical Diagnosis
  - Given a list of symptoms, predict whether a patient has disease X or not
- Weather
  - Based on temperature, humidity, etc… predict if it will rain tomorrow

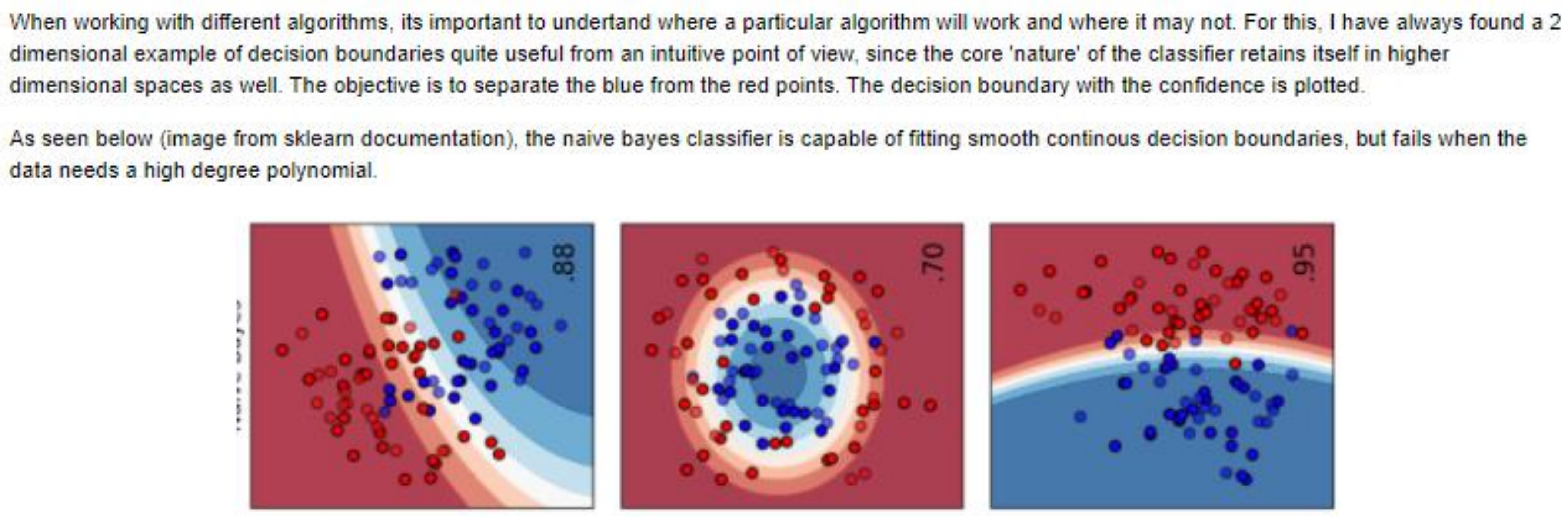
Types

Gaussian: It is used in classification, when the predictors are continuous and it assumes that features follow a normal/Gaussian distribution.

Multinomial: It is used for discrete counts and mostly used for document classification problems.

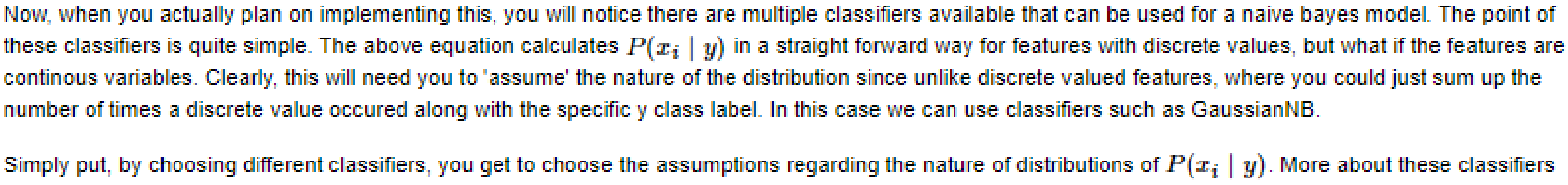
Bernoulli: similar to the multinomial naive bayes but the predictors are binary (i.e. zeros and ones).

Curve Fitting & intuition behing multiple classifiers



When working with different algorithms, its important to undertand where a particular algorithm will work and where it may not. For this, I have always found a 2 dimensional example of decision boundaries quite useful from an intuitive point of view, since the core 'nature' of the classifier retains itself in higher dimensional spaces as well. The objective is to separate the blue from the red points. The decision boundary with the confidence is plotted.

As seen below (image from sklearn documentation), the naive bayes classifier is capable of fitting smooth continous decision boundaries, but fails when the data needs a high degree polynomial.

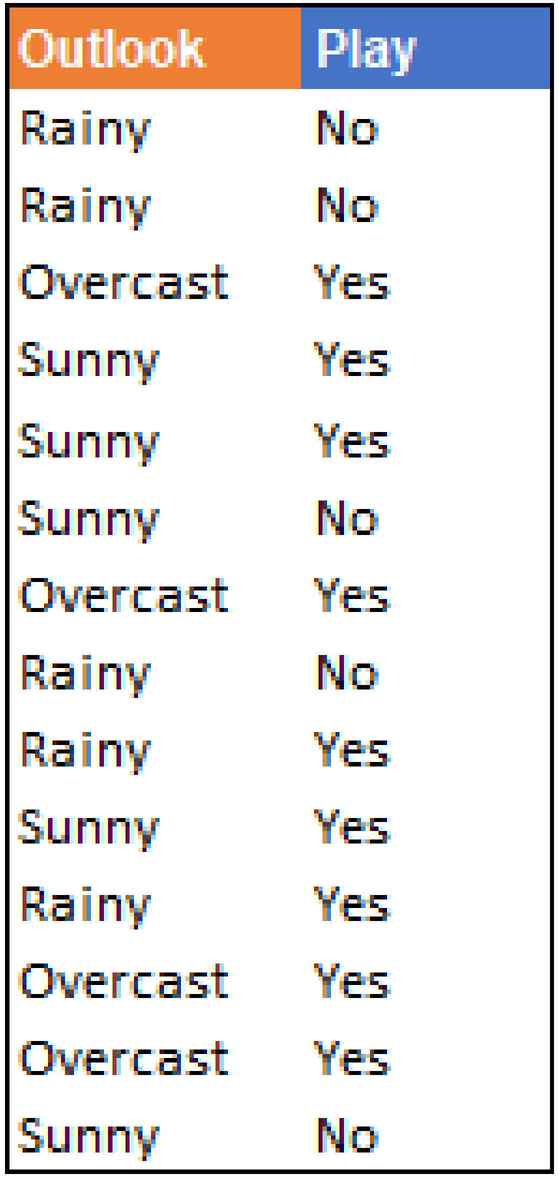


Now, when you actually plan on implementing this, you will notice there are multiple classifiers available that can be used for a naive bayes model. The point of these classifiers is quite simple. The above equation calculates $P(x_i \mid y)$ in a straight forward way for features with discrete values, but what if the features are continous variables. Clearly, this will need you to 'assume' the nature of the distribution since unlike discrete valued features, where you could just sum up the number of times a discrete value occured along with the specific y class label. In this case we can use classifiers such as GaussianNB.

Simply put, by choosing different classifiers, you get to choose the assumptions regarding the nature of distributions of $P(x_i \mid y)$. More about these classifiers

Reference: Akshay Sehgal on Naïve Bayes

Problem #1



Outlook	Play
Rainy	No
Rainy	No
Overcast	Yes
Sunny	Yes
Sunny	Yes
Sunny	No
Overcast	Yes
Rainy	No
Rainy	Yes
Sunny	Yes
Rainy	Yes
Overcast	Yes
Overcast	Yes
Sunny	No

Suppose we have a **Day** with the following values :

•Outlook = Sunny

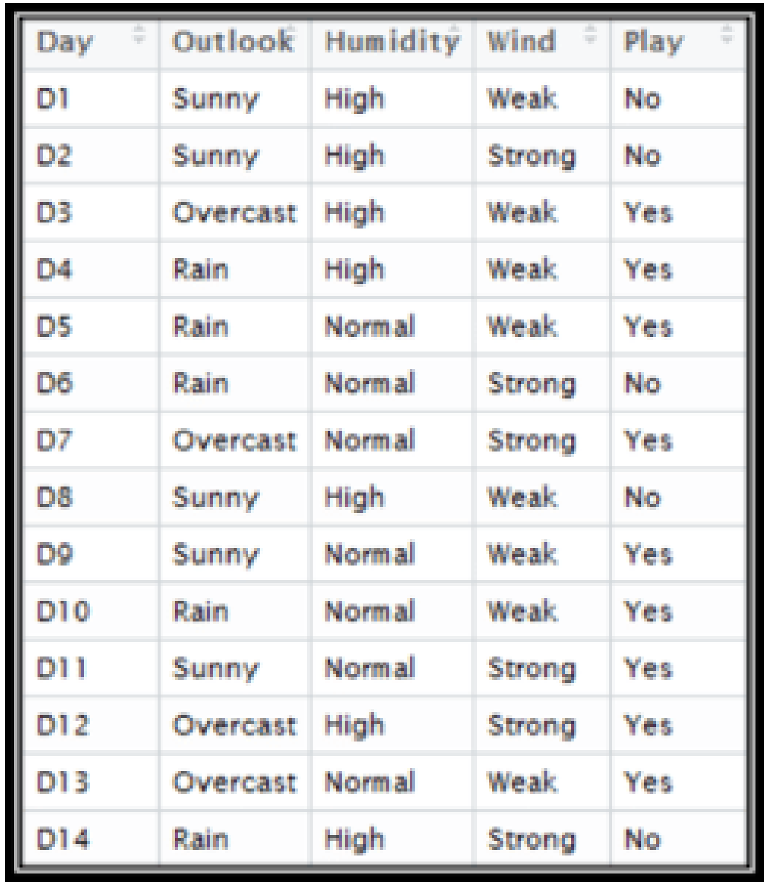
•Play = ?

Players will play given that if weather is sunny or not sunny ?

Give the answer based on NB

(refer to excel for solution)

Problem #2



Day	Outlook	Humidity	Wind	Play
D1	Sunny	High	Weak	No
D2	Sunny	High	Strong	No
D3	Overcast	High	Weak	Yes
D4	Rain	High	Weak	Yes
D5	Rain	Normal	Weak	Yes
D6	Rain	Normal	Strong	No
D7	Overcast	Normal	Strong	Yes
D8	Sunny	High	Weak	No
D9	Sunny	Normal	Weak	Yes
D10	Rain	Normal	Weak	Yes
D11	Sunny	Normal	Strong	Yes
D12	Overcast	High	Strong	Yes
D13	Overcast	Normal	Weak	Yes
D14	Rain	High	Strong	No

Let's predict the future with some weather data.

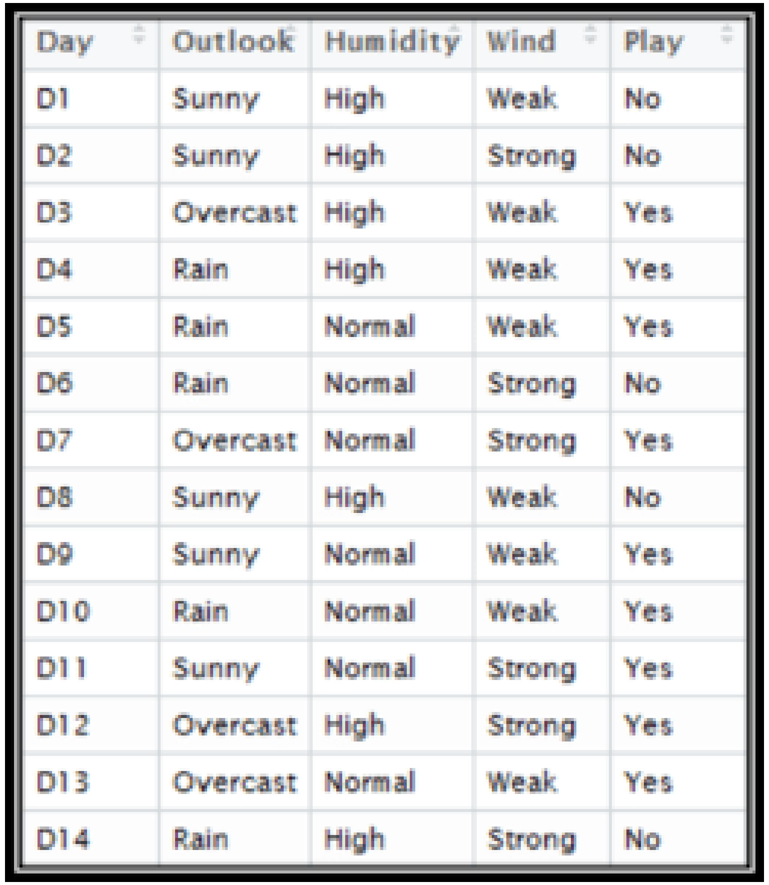
Here we have our data, which comprises the day, outlook, humidity, and wind conditions. The final column is 'Play,' i.e., can we play outside, which we have to predict.

Suppose we have a **Day** with the following values :

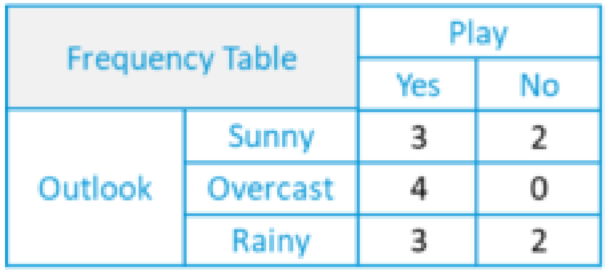
- Outlook = Rain
- Humidity = High
- Wind = Weak

- Play = ?

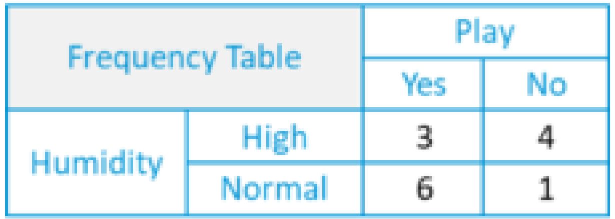
First, we will create a **frequency** table using each attribute of the dataset.



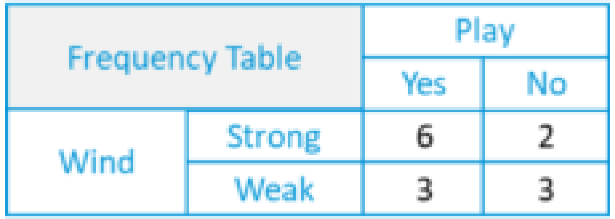
Day	Outlook	Humidity	Wind	Play
D1	Sunny	High	Weak	No
D2	Sunny	High	Strong	No
D3	Overcast	High	Weak	Yes
D4	Rain	High	Weak	Yes
D5	Rain	Normal	Weak	Yes
D6	Rain	Normal	Strong	No
D7	Overcast	Normal	Strong	Yes
D8	Sunny	High	Weak	No
D9	Sunny	Normal	Weak	Yes
D10	Rain	Normal	Weak	Yes
D11	Sunny	Normal	Strong	Yes
D12	Overcast	High	Strong	Yes
D13	Overcast	Normal	Weak	Yes
D14	Rain	High	Strong	No



Frequency Table		Play	
		Yes	No
Outlook	Sunny	3	2
	Overcast	4	0
	Rainy	3	2

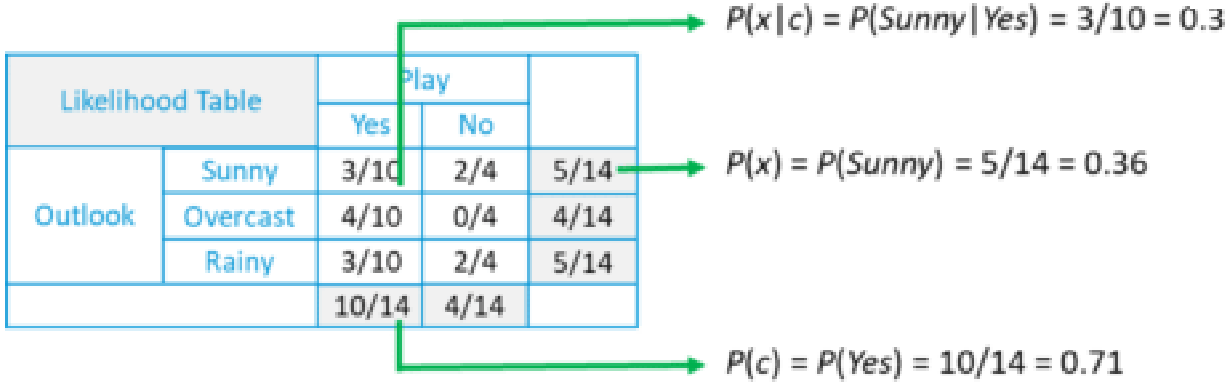


Frequency Table		Play	
		Yes	No
Humidity	High	3	4
	Normal	6	1



Frequency Table		Play	
		Yes	No
Wind	Strong	6	2
	Weak	3	3

For each frequency table, we will generate a **likelihood** table.



Likelihood Table		Play		
		Yes	No	
Outlook	Sunny	3/10	2/4	5/14
	Overcast	4/10	0/4	4/14
	Rainy	3/10	2/4	5/14
		10/14	4/14	

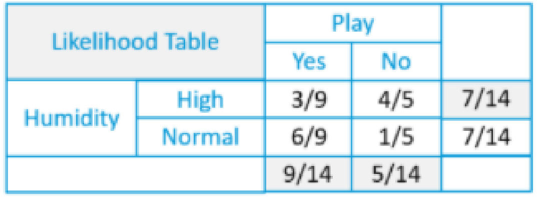
$P(x|c) = P(Sunny|Yes) = 3/10 = 0.3$

$P(x) = P(Sunny) = 5/14 = 0.36$

$P(c) = P(Yes) = 10/14 = 0.71$

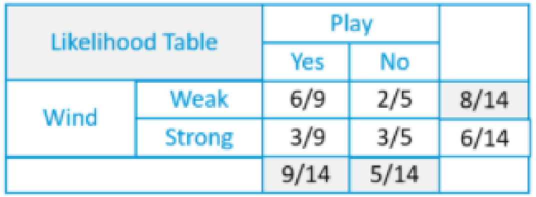
- Likelihood of '**Yes**' given '**Sunny**' is:
  - **P(c|x) = P(Yes|Sunny) = P(Sunny|Yes)* P(Yes) / P(Sunny) = (0.3 x 0.71) /0.36 = 0.591**
- Similarly, the likelihood of '**No**' given '**Sunny**' is:
  - **P(c|x) = P(No|Sunny) = P(Sunny|No)* P(No) / P(Sunny) = (0.4 x 0.36) /0.36 = 0.40**
- Now, in the same way, we need to create the Likelihood Table for other attributes as well.

Likelihood table for Humidity



Likelihood Table		Play		
		Yes	No	
Humidity	High	3/9	4/5	7/14
	Normal	6/9	1/5	7/14
		9/14	5/14	

Likelihood table for Wind



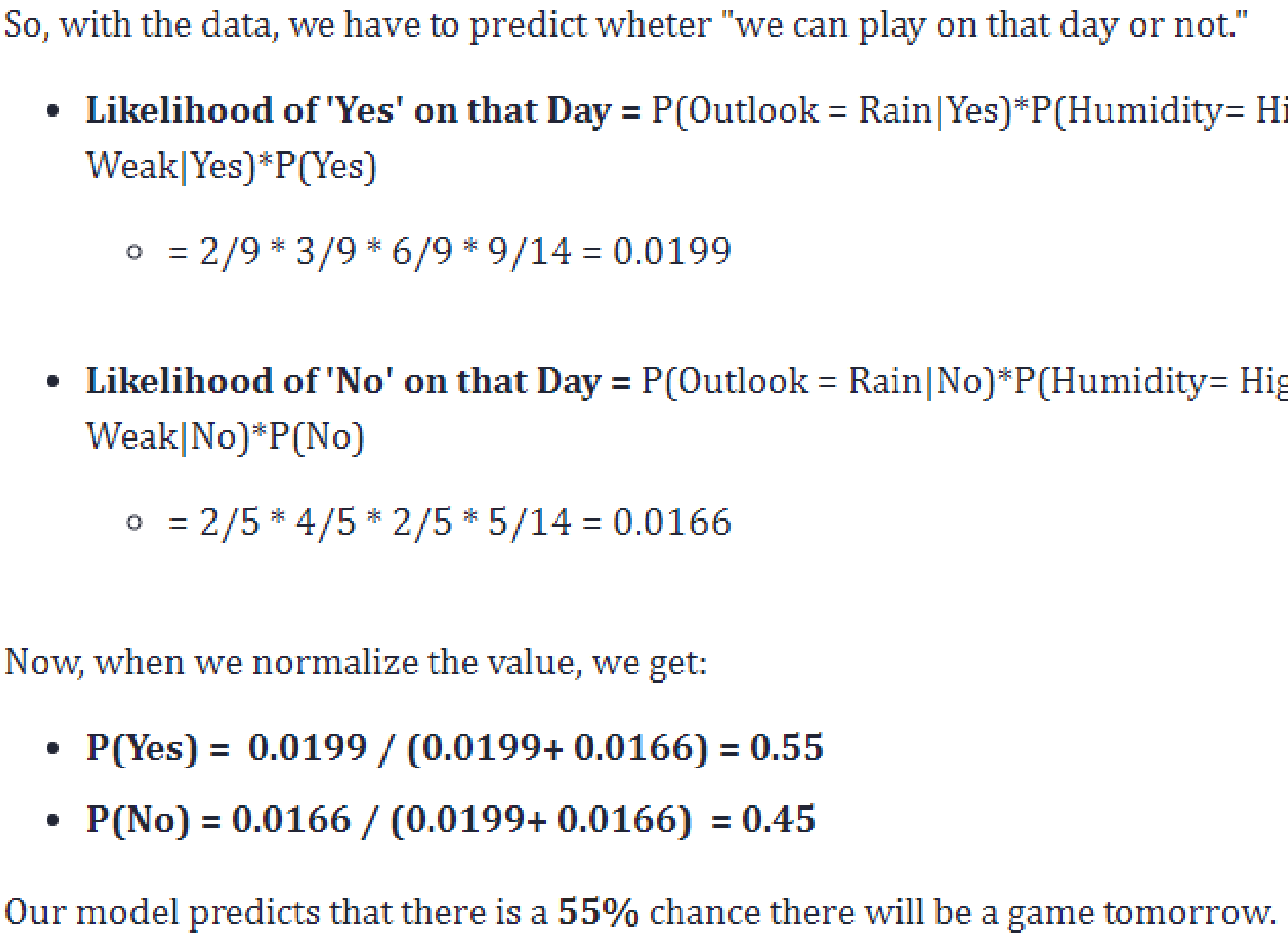
Likelihood Table		Play		
		Yes	No	
Wind	Weak	6/9	2/5	8/14
	Strong	3/9	3/5	6/14
		9/14	5/14	

$P(Yes|High) = 0.33 \times 0.6 / 0.5 = 0.42$

$P(No|High) = 0.8 \times 0.36 / 0.5 = 0.58$

$P(Yes|Weak) = 0.67 \times 0.64 / 0.57 = 0.75$

$P(No|Weak) = 0.4 \times 0.36 / 0.57 = 0.25$



So, with the data, we have to predict wheter "we can play on that day or not."

- **Likelihood of 'Yes' on that Day =** P(Outlook = Rain|Yes)*P(Humidity= High|Yes)* P(Wind= Weak|Yes)*P(Yes)
  - = 2/9 * 3/9 * 6/9 * 9/14 = 0.0199

- **Likelihood of 'No' on that Day =** P(Outlook = Rain|No)*P(Humidity= High|No)* P(Wind= Weak|No)*P(No)
  - = 2/5 * 4/5 * 2/5 * 5/14 = 0.0166

Now, when we normalize the value, we get:

- **P(Yes) = 0.0199 / (0.0199+ 0.0166) = 0.55**
- **P(No) = 0.0166 / (0.0199+ 0.0166) = 0.45**

Our model predicts that there is a **55%** chance there will be a game tomorrow.