LEANDRO MINKU

NAÏVE BAYES

Preliminaries

When discussing machine learning approaches that model probabilities and probability distributions, it is frequently useful to use a notation that more explicitly distinguishes between a variable and the value of this variable. This enables us to more clearly distinguish between probability distributions and the probabilities themselves. Therefore, when discussing Naïve Bayes, we will be using upper case letters such as X and Y to refer to independent and dependent variables, and lower case letters such as $\mathbf{a}$ and c to refer to values that these variables can assume.

The Relationship Between The Bayes Theorem and Classification

Supervised learning can be formulated as follows. Given a set of training examples:

$$\mathcal{T} = \{(\mathbf{a}^{(1)}, c^{(1)}), (\mathbf{a}^{(2)}, c^{(2)}), \cdots, (\mathbf{a}^{(N)}, c^{(N)})\},$$

where $(\mathbf{a}^{(i)}, c^{(i)}) \in \mathcal{X} \times \mathcal{Y}$ are drawn from a fixed albeit unknown joint probability distribution $p(\mathbf{X}, Y)$, $\mathcal{X}$ is the input space and $\mathcal{Y}$ is the output space.

Supervised learning consists in learning a model $f : \mathcal{X} \to \mathcal{Y}$ able to generalise to unseen examples of the same probability distribution $p(\mathbf{X}, Y)$. In regression problems, $\mathcal{Y} = \mathbb{R}$, whereas in classification problems $\mathcal{Y}$ is a set of categories.

The joint probability distribution $p(\mathbf{X}, Y)$ can be written as:

$$p(\mathbf{X}, Y) = p(Y|\mathbf{X})p(\mathbf{X}) = p(\mathbf{X}|Y)p(Y).$$

From the above, we have the following:

$$p(Y|\mathbf{X}) = \frac{p(Y)p(\mathbf{X}|Y)}{p(\mathbf{X})}.$$

Therefore, given an example with known value of $\mathbf{X} = \mathbf{a}$, one can calculate the probability of it belonging to a given class c as:

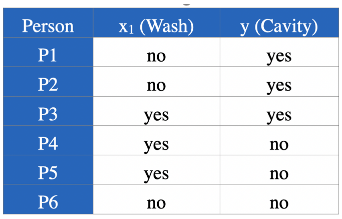
$$p(Y = c|\mathbf{X} = \mathbf{a}) = \frac{p(Y = c)p(\mathbf{X} = \mathbf{a}|Y = c)}{p(\mathbf{X} = \mathbf{a})}. \tag{1}$$

This is known as the Bayes Theorem. To simplify the writing, whenever it is not ambiguous, we will write this as:

$$p(c|\mathbf{a}) = \frac{p(c)(\mathbf{a}|c)}{p(\mathbf{a})}.$$

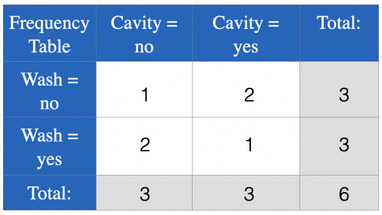
Such probabilities can be used to make inferences, i.e., to predict the class c given the observed input values $\mathbf{a}$. In particular, one can compute $p(c|\mathbf{a})$ for every possible class c, and predict the class associated to the highest probability $p(c|\mathbf{a})$. This means that these probabilities can work as our model $f : \mathcal{X} \to \mathcal{Y}$. But how to compute these probabilities?

In supervised learning, we could rely on the fact that we have access to a training set $\mathcal{T}$. This training set can be used to create frequency tables that enable us to compute these probabilities. Let's consider a simplified example where we have a single independent variable X_1 representing whether a person uses mouthwash and a dependent variable Y corresponding to whether that person has cavities. For this example, we will refer to X_1 as "Wash" and Y as "Cavity", to facilitate reading. Assume we have access to the training set corresponding to six people below:



Person	x₁ (Wash)	y (Cavity)
P1	no	yes
P2	no	yes
P3	yes	yes
P4	yes	no
P5	yes	no
P6	no	no

We can compute the number of times that each different value of the independent and dependent variables occur together as shown in the frequency table below:



Frequency Table	Cavity = no	Cavity = yes	Total:
Wash = no	1	2	3
Wash = yes	2	1	3
Total:	3	3	6

Then, let's say that we have received a new test instance corresponding to a person who uses mouthwash (i.e., Wash $=$ no) and wish to predict whether this person has cavities by applying the Bayes Theorem[1]. From Eq. 1, we have that:

$$p(\text{Cavity} = \text{yes}|\text{Wash} = \text{no}) =$$

$$\frac{p(\text{Cavity} = \text{yes})p(\text{Wash} = \text{no}|\text{Cavity} = \text{yes})}{p(\text{Wash} = \text{no})}$$

and

$$p(\text{Cavity} = \text{no}|\text{Wash} = \text{no}) =$$

$$\frac{p(\text{Cavity} = \text{no})p(\text{Wash} = \text{no}|\text{Cavity} = \text{no})}{p(\text{Wash} = \text{no})}$$

Based on the frequency table above, we have that:

[1] We could potentially compute $p(\text{Cavity} = \text{yes}|\text{Wash} = \text{no})$ and $p(\text{Cavity} = \text{no}|\text{Wash} = \text{no})$ directly from the frequency tables without having to apply the Bayes Theorem. However, the intention of this example is to show that we can compute them by applying the Bayes Theorem, which will be useful to learn Naïve Bayes in the next section.

- $p(\text{Cavity} = \text{yes}) = 3/6$, as 3 in 6 people had cavities.
- $p(\text{Wash} = \text{no}|\text{Cavity} = \text{yes}) = 2/3$, as 2 in 3 of the people with cavities did not use mouthwash.
- $p(\text{Cavity} = \text{no}) = 3/6$, as 3 in 6 people did not have cavities.
- $p(\text{Wash} = \text{no}|\text{Cavity} = \text{no}) = 1/3$, as 1 in 3 of the people with cavities used mouthwash.
- $p(\text{Wash} = \text{no}) = 3/6$, as 3 in 6 people did not use mouthwash.

Therefore,

$$p(\text{Cavity} = \text{yes}|\text{Wash} = \text{no}) = \frac{3/6 \times 2/3}{3/6} \approx 0.67.$$

$$p(\text{Cavity} = \text{no}|\text{Wash} = \text{no}) = \frac{3/6 \times 1/3}{3/6} \approx 0.33.$$

As $p(\text{Cavity} = \text{yes}|\text{Wash} = \text{no}) > p(\text{Cavity} = \text{no}|\text{Wash} = \text{no})$, it would be reasonable to predict that the class is Cavity = yes.
It is worth noting that the denominator $p(\mathbf{a})$ of the Bayes Theorem ($p(\text{Wash} = \text{no})$ in the example above) works as a normalisation factor to ensure that the probabilities of the test instance to belong to each different possible class sums to one, i.e.:

$$\sum_{c \in \mathcal{Y}} p(c|\mathbf{a}) = 1$$

We could replace $p(\mathbf{a})$ by the factor β shown below and achieve the same normalisation effect:

$$\beta = \sum_{c \in \mathcal{Y}} p(c)p(\mathbf{a}|c)$$

If we set $\alpha = 1/\beta$, then the Bayes Theorem becomes:

$$p(c|\mathbf{a}) = \alpha p(c)p(\mathbf{a}|c). \tag{2}$$

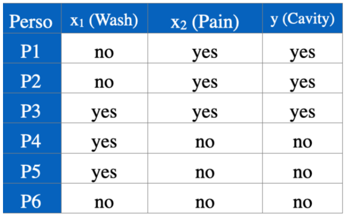
In the next section, it will be useful to explicitly use α instead of $p(\mathbf{a})$, as the assumptions made by Naïve Bayes will cause $p(\mathbf{a})$ not to work as a normalising factor anymore, potentially resulting in probabilities that do not sum to 1.

Naïve Bayes for Categorical Independent Variables

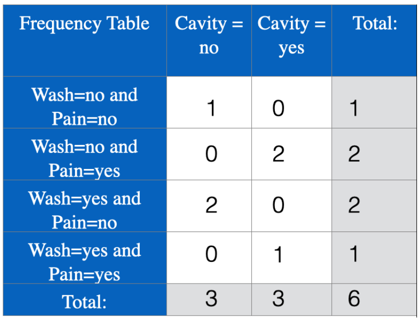
The Bayes Theorem can be read as follows for d independent variables:

$$p(c|\mathbf{a}) = \alpha p(c)p(\mathbf{a}|c) = \alpha p(c)p(a_1, a_2, \cdots, a_d|c).$$

As the probability $p(a_1, a_2, \cdots, a_d|c)$ corresponds to the probability of a given combination of values $a_1, a_2, \cdots, a_d$ for the independent variables being observed together given a class c, each row of our frequency table would correspond to a different combination of values for the independent variables. For example, for a problem with two independent variables, we could have the training set and frequency table below:



Perso	x_1 (Wash)	x_2 (Pain)	y (Cavity)
P1	no	yes	yes
P2	no	yes	yes
P3	yes	yes	yes
P4	yes	no	no
P5	yes	no	no
P6	no	no	no



Frequency Table	Cavity = no	Cavity = yes	Total:
Wash=no and Pain=no	1	0	1
Wash=no and Pain=yes	0	2	2
Wash=yes and Pain=no	2	0	2
Wash=yes and Pain=yes	0	1	1
Total:	3	3	6

For problems with large numbers of independent variables and many possible values for such variables, the number of rows in the frequency table would quickly become intractable.

Naïve Bayes is a machine learning approach that deals with this issue by making the assumption that the input variables are conditionally independent of each other given the dependent variable. A variable X_1 is conditionally independent of X_2 given the dependent variable Y if the following condition is satisfied:

$$p(X_1, X_2|Y) = p(X_1|Y)$$

This means that, if we know the value of Y, we do not need to know the value of X_2 in order to determine the value of X_1.

By making this assumption, we can say that:

$$P(a_1, a_2, \cdots, a_d|c) = \prod_{i=1}^{d} p(a_i|c).$$

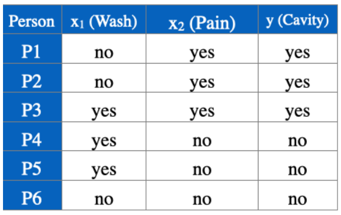
By replacing this in the Bayes Theorem (Equation 2), we get:

$$p(c|\mathbf{a}) = \alpha p(c) \prod_{i=1}^{d} p(a_i|c). \qquad (3)$$

where $\alpha = \sum_{c \in \mathcal{Y}} \left(p(c) \prod_{i=1}^{d} p(a_i|c) \right)$

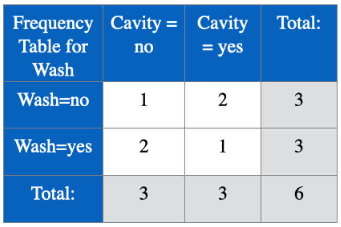
Naïve Bayes uses the equation above to calculate $p(c|\mathbf{a})$ in order to predict the class associated to a given example whose independent variables have value $\mathbf{a}$. This means that a separate frequency table can be created for each independent variable, leading to a total number of rows that grows linearly with the number of values that the independent variables can assume. An example is shown below.

Let's consider a problem where we wish to predict whether a person has cavity based on whether they use mouthwash and are experiencing tooth pain. The training set for this problem is shown below:

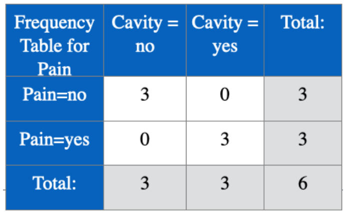


Person	x_1 (Wash)	x_2 (Pain)	y (Cavity)
P1	no	yes	yes
P2	no	yes	yes
P3	yes	yes	yes
P4	yes	no	no
P5	yes	no	no
P6	no	no	no

The frequency tables corresponding to the training set above are shown below:



Frequency Table for Wash	Cavity = no	Cavity = yes	Total:
Wash=no	1	2	3
Wash=yes	2	1	3
Total:	3	3	6



Frequency Table for Pain	Cavity = no	Cavity = yes	Total:
Pain=no	3	0	3
Pain=yes	0	3	3
Total:	3	3	6

Let's assume that we wish to predict whether a person who uses mouthwash (Wash = yes) and is experiencing tooth pain (Pain = yes) has a cavity. We would need to compute the following:

$$p(\text{Cavity} = \text{yes}|\text{Wash} = \text{yes}, \text{Pain} = \text{yes}) =$$

$$\alpha p(\text{Cavity} = \text{yes}) \times p(\text{Wash} = yes|\text{Cavity} = \text{yes}) \times p(\text{Pain} = yes|\text{Cavity} = \text{yes}) =$$

$$\alpha \times 3/6 \times 1/3 \times 3/3 = \alpha \times 1/6$$

$$p(\text{Cavity} = \text{no}|\text{Wash} = \text{yes}, \text{Pain} = \text{yes}) =$$

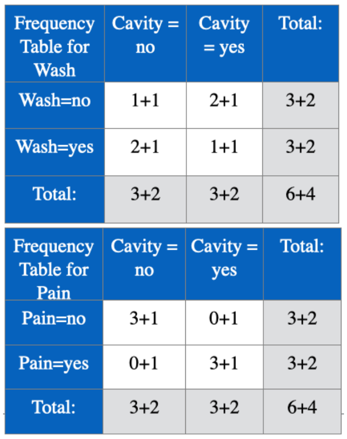
$$\alpha p(\text{Cavity} = \text{no}) \times p(\text{Wash} = yes|\text{Cavity} = \text{no}) \times p(\text{Pain} = yes|\text{Cavity} = \text{no}) =$$

$$\alpha \times 3/6 \times 2/3 \times 0/3 = 0$$

where $\alpha = 3/6 \times 1/3 \times 3/3 + 3/6 \times 2/3 \times 0/3 = 1/6$.

As $p(\text{Cavity} = \text{yes}|\text{Wash} = \text{yes}, \text{Pain} = \text{yes}) > p(\text{Cavity} = \text{no}|\text{Wash} = \text{yes}, \text{Pain} = \text{yes})$, Naïve Bayes predicts that the person has cavity.

Note that in the calculations above, the whole probability of the person not having cavity became zero, because $p(\text{Pain} = yes|\text{Cavity} = \text{no}) = 0$. In fact, even if there had been many other independent variables in this problem, all with values suggesting that this person did not have cavity, the whole probability of the person not having cavity would still become zero, just because $p(\text{Pain} = \mathit{yes}|\text{Cavity} = \text{no}) = 0$. This would lead to several inaccuracies on Naïve Bayes' predictions. One way to avoid this problem is to adopt Laplace Smoothing. This consists in summing one to every cell in the frequency table, except for the total values cells, which would be updated according to the total values in the rows and columns of the table. And example of that is shown below:



Frequency Table for Wash	Cavity = no	Cavity = yes	Total:
Wash=no	1+1	2+1	3+2
Wash=yes	2+1	1+1	3+2
Total:	3+2	3+2	6+4

Frequency Table for Pain	Cavity = no	Cavity = yes	Total:
Pain=no	3+1	0+1	3+2
Pain=yes	0+1	3+1	3+2
Total:	3+2	3+2	6+4

When computing the probabilities for Equation 3, the values of $p(a_i|c)$ should be computed based on the updated tables, whereas $p(c)$ is still calculated based on the original tables (without Laplace Smoothing). For instance, the calculations to predict whether a person who uses mouthwash and is experiencing tooth pain would be as follows:

$$p(\text{Cavity} = \text{yes}|\text{Wash} = \text{yes}, \text{Pain} = \text{yes}) =$$

$$\alpha p(\text{Cavity} = \text{yes}) \times p(\text{Wash} = \mathit{yes}|\text{Cavity} = \text{yes}) \times p(\text{Pain} = \mathit{yes}|\text{Cavity} = \text{yes}) =$$

$$\alpha \times 3/6 \times 2/5 \times 4/5 = \alpha \times 8/50$$

$$p(\text{Cavity} = \text{no}|\text{Wash} = \text{yes}, \text{Pain} = \text{yes}) =$$

$$\alpha p(\text{Cavity} = \text{no}) \times p(\text{Wash} = \mathit{yes}|\text{Cavity} = \text{no}) \times p(\text{Pain} = \mathit{yes}|\text{Cavity} = \text{no}) =$$

$$\alpha \times 3/6 \times 3/5 \times 1/5 = 3/50$$

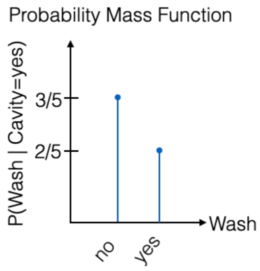
where $\alpha = 3/6 \times 2/5 \times 4/5 + 3/6 \times 3/5 \times 1/5 = 8/50 + 3/50$

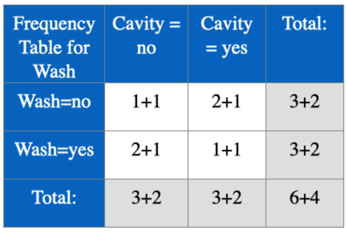
As $p(\text{Cavity} = \text{yes}|\text{Wash} = \text{yes}, \text{Pain} = \text{yes}) > p(\text{Cavity} = \text{no}|\text{Wash} = \text{yes}, \text{Pain} = \text{yes})$, Naïve Bayes still predicts that the person has cavity in this example. However, in other examples the predicted class could potentially change when adopting Laplace Smoothing.

Naïve Bayes for Numeric Independent Variables

The Naïve Bayes frequency tables explained in the previous section contain one row for each possible value of the independent variables. When dealing with numeric independent variables, it is infeasible to have one row for each possible numeric value. How can Naïve Bayes deal with numeric independent variables?

To gain some insight into that, it is worth observing that when we are collecting frequency values for the frequency tables and transforming them into probabilities, we are actually learning probability mass functions. For instance, the figure below shows an example of probability mass function for $p(\text{Wash}|\text{Cavity} = \text{yes})$.





Frequency Table for Wash	Cavity = no	Cavity = yes	Total:
Wash=no	1+1	2+1	3+2
Wash=yes	2+1	1+1	3+2
Total:	3+2	3+2	6+4

For numeric independent variables, we can learn probability density functions, instead of probability mass functions. Probability density functions represent the relative likelihood of observing different values of a given numeric variable. Different from probability mass functions, the values of the relative likelihoods do not sum to one, as explained in Iain Style's Math notes on Random Variables. Instead, the area under the curve formed by the function is equal to one.

To be able to adopt probability density functions in Naïve Bayes, one must choose what kind of probability density function to adopt. In most cases, a univariate Gaussian probability density function is used for each numeric independent variable. This function can be written as follows for a given numeric independent variable X:

$$p(X|\mu, \sigma^2) = \frac{1}{\sqrt{2\sigma^2\pi}} e^{\frac{-(X-\mu)^2}{2\sigma^2}}$$

where μ is a parameter corresponding to the mean of the distribution, σ^2 is a parameter corresponding to the variance, $\pi \approx 3.14159$ and $e \approx 2.71828$.

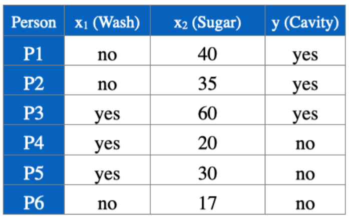
Learning then corresponds to setting appropriate values for the parameters μ and σ^2. In classification problems, every numeric independent variable X needs to be associated to a probability density function $p(X|c)$ for each of the possible values $c \in \mathcal{Y}$. In order to learn the parameters of these probability density functions, the training examples can be used. In particular, for a Gaussian probability density function associated to the independent variable X and the class value c, the mean μ is set as the mean of the X values values of all training examples whose class value is c. The variance σ^2 is set as the variance of these values. The mean and variance formulas for a set of values V are shown below:

$$\mu(V) = \frac{1}{|V|} \sum_{v \in V} v$$

$$\sigma^2(V) = \frac{1}{|V| - 1} \sum_{v \in V} (v - \mu(V))^2$$

where $|V|$ is the number of elements in V. An example is shown below.

Let's assume that we have a problem where we need to predict whether a person has a cavity (dependent variable Y) based on whether this person uses mouthwash (categorical independent variable X_1) and on the amount of sugar in grams that this person consumes per day (numeric independent variable X_2). The training set is shown below:



Person	x_1 (Wash)	x_2 (Sugar)	y (Cavity)
P1	no	40	yes
P2	no	35	yes
P3	yes	60	yes
P4	yes	20	no
P5	yes	30	no
P6	no	17	no

Consider that we decide to use Gaussian probability density functions for the numeric independent variable Sugar (X_2). As this is a binary classification problem (where $\mathcal{Y} = \{\text{yes}, \text{no}\}$ is a set of size 2), this means that we need to learn two Gaussian probability density functions for Sugar – one for Cavity $=$ yes and one for Cavity $=$ no.

For Cavity $=$ yes, we calculate the mean and variance of the Sugar values of all training examples where Cavity $=$ yes:

$$\mu = \frac{40 + 35 + 60}{3} = 45$$

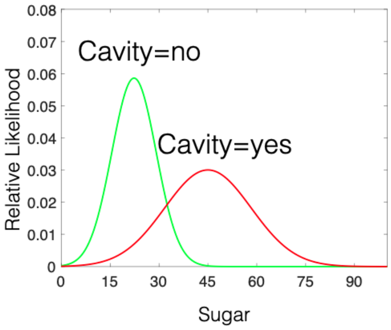
$$\sigma^2 = \frac{1}{3-1}[(40-45)^2 + (35-45)^2 + (60-45)^2] = 175$$

For Cavity $=$ no, we calculate the mean and variance of the Sugar values of all training examples where Cavity $=$ no:

$$\mu = \frac{20 + 30 + 17}{3} \approx 22.33$$

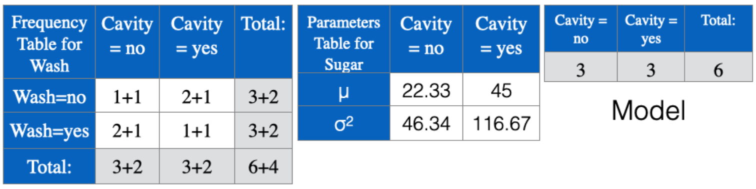
$$\sigma^2 = \frac{1}{3-1}[(20-22.33)^2 + (30-22.33)^2 + (17-22.33)^2] \approx 46.34$$

Therefore, one of the Gaussian probability density functions is $p(\text{Sugar}|\mu = 45, \sigma^2 = 175)$ and the other is $p(\text{Sugar}|\mu = 22.33, \sigma^2 = 46.34)$. Note that we are omitting Cavity $=$ yes and Cavity $=$ no when we write $p(\text{Sugar}|\mu = 45, \sigma^2 = 175)$ and $p(\text{Sugar}|\mu = 22.33, \sigma^2 = 46.34)$ because the μ and σ^2 values already capture Cavity $=$ yes and Cavity $=$ no. However, we could also write Cavity $=$ yes and Cavity $=$ no explicitly if we wanted. These two functions are plotted below:



Once the parameters of the probability density functions are set, the values of $p(a_i|c)$ used in Equation 3 can be taken from these functions. It's worth noting that these values are relative likelihood instead of probabilities. However, the resulting $p(c|a_i)$ will still be probabilities due to the use of the normalising factor α.

An example of prediction for this problem would be as follows. Consider that we wish to predict whether a person who does not use mouthwash (Wash $=$ no) and consumes 20 grams of sugar per day (Sugar $=$ 20) has cavity. The frequency tables with Laplace smoothing, the total number of training examples for Cavity $=$ yes and Cavity $=$ no and the parameters of the probability density functions computed based on the training set are shown below.



Frequency Table for Wash	Cavity = no	Cavity = yes	Total:
Wash=no	1+1	2+1	3+2
Wash=yes	2+1	1+1	3+2
Total:	3+2	3+2	6+4

Parameters Table for Sugar	Cavity = no	Cavity = yes
μ	22.33	45
σ2	46.34	116.67

Cavity = no	Cavity = yes	Total:
3	3	6

Model

The probabilities $p(\text{Cavity} = \text{yes}|\text{Wash} = \text{no}, \text{Sugar} = 20)$ and $p(\text{Cavity} = \text{no}|\text{Wash} = \text{no}, \text{Sugar} = 20)$ are shown below:

$$p(\text{Cavity} = \text{yes}|\text{Wash} = \text{no}, \text{Sugar} = 20) =$$

$$\alpha p(\text{Cavity} = \text{yes}) p(\text{Wash} = \text{no}|\text{Cavity} = \text{yes}) p(\text{Sugar} = 20|\text{Cavity} = \text{yes}) =$$

$$\alpha \times 3/6 \times 3/5 \times 0.0051 = \alpha \times 0.00153 = 12.15\%$$

$$\begin{gathered}
p(\text{Cavity} = \text{no}|\text{Wash} = \text{no}, \text{Sugar} = 20) = \\
\alpha p(\text{Cavity} = \text{no}) p(\text{Wash} = \text{no}|\text{Cavity} = \text{no}) p(\text{Sugar} = 20|\text{Cavity} = \text{no}) = \\
\alpha \times 3/6 \times 2/5 \times 0.0553 = \alpha \times 0.01106 = 87.85\%
\end{gathered}$$

where $\alpha = 1/(0.00153 + 0.01106) \approx 79.43$

Advantages, Disadvantages and Applications of Naïve Bayes

Naïve Bayes' advantages include:

- Training is fast, as it requires only one pass through the data.
- The relative probabilities computed by Naïve Bayes are good for making predictions for many applications, such as text categorisation (e.g., spam detection), medical diagnosis, software defect prediction, among others.

Naïve Bayes' disadvantages include:

- It assumes conditional independence.
- It assumes a certain probability distribution for numeric independent variables.
- It does not work very well for regression problems, despite variations of it being available for this type of problems.

Reading

Bishop's book "Machine Learning and Pattern Recognition", pages 45-46 (Combining Models).

Russell and Norvig's "Artificial Intelligence: A Modern Approach", Sections 13.2 (Basic Probability Notation) to 13.6(The Wumpus World Revisited).