First, let's define abbreviations for some logical assertions:

- $f_i$: The message contains feature $i$.
- $\neg f_j$: The message does not contain feature $j$.
- ham: The message is ham.
- spam: The message is spam.

We wish to compute:

\[
P(\text{ham} \mid f_1, \ldots, f_m, \neg f_{m+1}, \ldots, \neg f_n)\]

That is, the probability that the message is ham, given that the message contains some features and does not contain others.

\[
P(\text{spam} \mid f_1, \ldots, f_m, \neg f_{m+1}, \ldots, \neg f_n)\]

That is, the probability that the message is spam, given that the message contains some features and does not contain others.

That notation is cumbersome. So let's abbreviate "$f_1, \ldots, f_m, \neg f_{m+1}, \ldots, \neg f_n$" as "$f_1, \ldots, \neg f_n$". So we wish to compute:

\[
P(\text{ham} \mid f_1, \ldots, \neg f_n)\]

\[
P(\text{spam} \mid f_1, \ldots, \neg f_n)\]

But we don't know how to compute those probabilities. So let's apply some mathematics...
Since the AND operator is commutative:
\[ P(f_1, \ldots, \neg f_n \text{ AND } \text{ham}) = P(\text{ham AND } f_1, \ldots, \neg f_n) \]

By the multiplicative law of probability, \( P(x \text{ AND } y) = P(x) \ P(y \mid x) \). And so:
\[ P(f_1, \ldots, \neg f_n) \ P(\text{ham} \mid f_1, \ldots, \neg f_n) = P(\text{ham}) \ P(f_1, \ldots, \neg f_n \mid \text{ham}) \]

Dividing both sides of the equation by \( P(f_1, \ldots, \neg f_n) \), we get Bayes' Rule:
\[ P(\text{ham} \mid f_1, \ldots, \neg f_n) = \frac{P(\text{ham}) \ P(f_1, \ldots, \neg f_n \mid \text{ham})}{P(f_1, \ldots, \neg f_n)} \]

Similarly:
\[ P(f_1, \ldots, \neg f_n \text{ AND spam}) = P(\text{spam AND } f_1, \ldots, \neg f_n) \]
\[ P(f_1, \ldots, \neg f_n) \ P(\text{spam} \mid f_1, \ldots, \neg f_n) = P(\text{spam}) \ P(f_1, \ldots, \neg f_n \mid \text{spam}) \]
\[ P(\text{spam} \mid f_1, \ldots, \neg f_n) = \frac{P(\text{spam}) \ P(f_1, \ldots, \neg f_n \mid \text{spam})}{P(f_1, \ldots, \neg f_n)} \]

Substituting for the above expressions, we wish to compute:

\[
\begin{array}{c}
P(\text{ham}) \ P(f_1, \ldots, \neg f_n \mid \text{ham}) \\
\hline
P(f_1, \ldots, \neg f_n)
\end{array}
\]

\[
\begin{array}{c}
P(\text{spam}) \ P(f_1, \ldots, \neg f_n \mid \text{spam}) \\
\hline
P(f_1, \ldots, \neg f_n)
\end{array}
\]

We don't know \( P(\text{ham}) \) or \( P(\text{spam}) \), and we never will. We'll need to guess them, based upon our perception of the proportion of our e-mail that is, in fact, spam.

\( P(f_1, \ldots, \neg f_n \mid \text{ham}) \) and \( P(f_1, \ldots, \neg f_n \mid \text{spam}) \) would be difficult to compute. Doing so would be possible only if we have examples of messages that contain (or do not contain) every combination of features. That would imply that we need a very large number of examples.

So we must make a simplifying assumption. Let's assume that \( f_1, \ldots, f_n \) are independent. Then:
\[ P(f_1, \ldots, \neg f_n) = P(f_1) \ldots P(\neg f_n) \]

and

\[
\begin{align*}
P(f_1, \ldots, \neg f_n \mid \text{ham}) &= P(f_1 \mid \text{ham}) \ldots P(\neg f_n \mid \text{ham}) \\
P(f_1, \ldots, \neg f_n \mid \text{spam}) &= P(f_1 \mid \text{spam}) \ldots P(\neg f_n \mid \text{spam})
\end{align*}
\]

Substituting into expressions (3), we wish to compute:

\[
\frac{P(\text{ham}) \ P(f_1 \mid \text{ham}) \ldots P(\neg f_n \mid \text{ham})}{P(f_1) \ldots P(\neg f_n)} + \frac{P(\text{spam}) \ P(f_1 \mid \text{spam}) \ldots P(\neg f_n \mid \text{spam})}{P(f_1) \ldots P(\neg f_n)} = 1
\]

We don't know how to compute \( P(f_i) \) or \( P(\neg f_i) \) for any \( i \). So we need to transform the denominators.

Certainly, any message is either ham or spam. So:

\[ P(\text{ham} \mid f_1, \ldots, \neg f_n) + P(\text{spam} \mid f_1, \ldots, \neg f_n) = 1 \]

After substituting expressions (4) into that equation:

\[
\frac{P(\text{ham}) \ P(f_1 \mid \text{ham}) \ldots P(\neg f_n \mid \text{ham})}{P(f_1) \ldots P(\neg f_n)} + \frac{P(\text{spam}) \ P(f_1 \mid \text{spam}) \ldots P(\neg f_n \mid \text{spam})}{P(f_1) \ldots P(\neg f_n)} = 1
\]

After multiplying both sides of the equation by \( P(f_1) \ldots P(\neg f_n) \):

\[ P(f_1) \ldots P(\neg f_n) = P(\text{ham}) \ P(f_1 \mid \text{ham}) \ldots P(\neg f_n \mid \text{ham}) + P(\text{spam}) \ P(f_1 \mid \text{spam}) \ldots P(\neg f_n \mid \text{spam}) \]

Substituting into expressions (4), we wish to compute:
As noted previously, the sum of those two expressions is 1. So it would be sufficient to compute only one of them; we easily could derive the other. So, let's compute only the second expression:

Note that we can estimate all components of that expression. Specifically, we can estimate:

- \( P(ham) \) based upon our perception of the proportion of our e-mail that is, in fact, ham.
- \( P(spam) \) based upon our perception of the proportion of our e-mail that is, in fact, spam. Note that \( P(ham) + P(spam) = 1 \).
- \( P(f_i \mid ham) \) by examining many ham messages, and determining the proportion of them that contain feature \( f_i \).
- \( P(f_i \mid spam) \) by examining many spam messages, and determining the proportion of them that contain feature \( f_i \).
- \( P(\neg f_i \mid ham) \) by examining many ham messages, and determining the proportion of them that do not contain feature \( f_i \). Or we could compute it as \( 1 - P(f_i \mid ham) \).
- \( P(\neg f_i \mid spam) \) by examining many spam messages, and determining the proportion of them that do not contain feature \( f_i \). Or we could compute it as \( 1 - P(f_i \mid spam) \).

So, in theory, we can use expression (6) to produce the results that we wish.

However, in practice the products may become very small (i.e. close to 0), and thus cause may cause floating-point underflow. So, relying upon the equality:
\[ x \cdot y = \exp(\log(x) + \log(y)) \]

Let's compute sums of logarithms instead of products. That is, we wish to compute:

\[
\begin{align*}
& \exp(\log(P(\text{spam})) + \log(P(f_1|\text{spam})) + \ldots + \log(P(-f_n|\text{spam}))) \\
& \quad + \exp(\log(P(\text{ham})) + \log(P(f_1|\text{ham})) + \ldots + \log(P(-f_n|\text{ham})))
\end{align*}
\]

The logarithms will be negative. So the sums of the logarithms may be large negative numbers (i.e. far from 0). So applying the exp operation to those sums may cause precisely the same floating-point underflow that motivated us to use logarithms in the first place.

Relying upon this equality:

\[
\frac{\exp(\log(x))}{\exp(\log(x) + \log(y))} = \frac{\exp(\log(x) + k)}{\exp(\log(x) + k) + \exp(\log(y) + k)}
\]

Let's add some number \( k \) to each sum-of-logs before applying the exp operator. So we wish to compute:

\[
\begin{align*}
& \exp(\log(P(\text{spam})) + \log(P(f_1|\text{spam})) + \ldots + \log(P(-f_n|\text{spam}))) + k \\
& \quad + \exp(\log(P(\text{ham})) + \log(P(f_1|\text{ham})) + \ldots + \log(P(-f_n|\text{ham}))) + k
\end{align*}
\]

For \( k \), a good choice would be:

\[-(\log(P(\text{spam})) + \log(P(f_1|\text{spam})) + \ldots + \log(P(-f_n|\text{spam})))
\]

That is,

\[-\log(P(\text{spam})) - \log(P(f_1|\text{spam})) - \ldots - \log(P(-f_n|\text{spam}))\]

It's a good choice because it makes the numerator equal \( \exp(0) \), that is, 1. It also makes the second term of the denominator equal \( \exp(0) \), that is, 1. So, we wish to compute:
Note that only one sum-of-logs remains, and it will not evaluate to a large negative number (i.e. far from 0). So applying the exp operation to that sum will not cause floating-point underflow. Thus we have an expression that is correct in theory and in practice.