1 Naive Bayes algorithm

Naive Bayes algorithm is quite well described and documented throughout the Internet, yet it is a good exercise for me to provide my own description. So, at first answer to few base questions. What is it about? This algorithm is classification technique, which means that based on some input data it assignes given category for each entry. We can give some simple examples, like determining gender or spam email based on data delivered. But we can also imagine, that we want to assign clients to particular segments and want similar clients to land in the same bucket. Ok, but why naive? Well, in an example we will notice that every input factor is treated independently and with same weight - so approach as simple as possible.
Last thing, almost only to give me chance to pratice writing math equations. Algorithm uses conditional probability, which is given by following equation:

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

We are not going to explain here its nature, but rather see on an example, how it works. On purpose, I will not present further any math equations.
Important: example given shows binary classification problem, where we assign element to one ouf of two categories, however approach for more categories works exactly the same way.
The most famous "problem" that is presented, when talking about this algorithm, is golf example. Therefore here I will give a different one. At first we need input set, with data, from which the algorithm will "learn".

Heigth	**Weigth**	**Eye color**	**Result**
Tall	Skinny	Blue	Pass
Tall	Medium	Brown	Pass
Short	Fat	Brown	Fail
Medium	Medium	Green	Fail
Short	Skinny	Blue	Pass
Medium	Fat	Blue	Fail
Tall	Fat	Green	Fail
Tall	Fat	Blue	Pass
Medium	Skinny	Green	Pass
Medium	Skinny	Brown	Fail

Ok, so here we have candidates that were trying to pass physical exams to a military university. Each of candidates is described by three factors: heigth, weigth and eye color. We also know, whether each of them passed or failed

the exam.
If you notice, this set was determined on purpose this way, cause there is one condition that is satisfied here. Namely, for each factor we can find entry with both results Pass and Fail. Later we will mention, what happens in cases when it is not fulfilled.
Neverthless, for now let us go forward with this table. First part of algorithm is about to collect each factor separately. So we have:

Heigth	**Pass**	**Fail**	Pr_{pass}	Pr_{fail}
Short	1	1	1/5	1/5
Medium	1	3	1/5	3/5
Tall	3	1	3/5	1/5
Total	**5**	**5**	**5/5**	**5/5**

So, you can see that we take a single factor, for each value we check how many failes and passes we have. Then, we calculate probability, e.g. for *Short* we have one pass, so probability is calculated as this one entry divided by all entries available where we have Pass, so five.
Overall we can notice that sum of these probabilities equals 100%.
Well, let us then build similar summary for two other factors.

Weigth	**Pass**	**Fail**	Pr_{pass}	Pr_{fail}
Skinny	3	1	3/5	1/5
Medium	1	1	1/5	1/5
Fat	1	3	1/5	3/5

Eye color	**Pass**	**Fail**	Pr_{pass}	Pr_{fail}
Blue	3	1	3/5	1/5
Brown	1	2	1/5	2/5
Green	1	2	1/5	2/5

In second step, we build same collation, but for our result. This is the value, which we will try to determine for future cases.

Result	**Number**	**Pr**
Pass	5	5/10
Fail	5	5/10

Now, using this information, we can try to calculate probability for a new student, whether he will pass or fail. Let's take following example: (Short, Medium, Green). We perform two calculations, probability that he passes and that he fails. For pass, we take probabilities for each factor:

Pr of pass under condition of being Short = 1/5
Pr of pass under condition of being Medium = 1/5
Pr of pass under condition of having Green eyes = 1/5
Pr of pass in general = 5/10
We multiply all these values getting as final result $P_{pass} = 1/5 * 1/5 * 1/5 * 5/10 = 1/250 = 0.004$

We do similar operation for fail, namely:
Pr of fail under condition of being Short = 1/5
Pr of fail under condition of being Medium = 1/5
Pr of fail under condition of having Green eyes = 2/5
Pr of fail in general = 5/10
Our final result is then $P_{fail} = 1/5 * 1/5 * 2/5 * 5/10 = 2/250 = 0.008$

Now, we know that total probability equals 1, so we can normalize these two values, by diving each of them by their sum:
$P_{pass} = \frac{0.004}{0.004+0.008} = 0.33$
$P_{fail} = \frac{0.008}{0.004+0.008} = 0.67$
We can then see, that for this student we have 2/3 chances that he will fail the exam and 1/3 that he will pass it (inidicating final prediction as Fail). This example shows very nicely naitivity of the algorithm. As you can observe, for parameters of heigth and weigth we had same probability of pass and fail. Therefore, the only factor that had actual influence to a final result, was eye color, which should not actually be so important when we are saying about passing physical exams. However, this algorithm treats all factors equally, as it was mentioned earlier.

Zero frequency problem
As I said before, with this example we have all values nicely present, but what if we had a zero in one of our probability matrix? Let us reduce example set by removing 9^{th} entry. Our frequency matrices will look then as follows:

Heigth	**Pass**	**Fail**	Pr_{pass}	Pr_{fail}
Short	1	1	1/4	1/5
Medium	0	3	0/4	3/5
Tall	3	1	3/4	1/5

Weigth	**Pass**	**Fail**	Pr_{pass}	Pr_{fail}
Skinny	2	1	2/4	1/5
Medium	1	1	1/4	1/5
Fat	1	3	1/4	3/5

Eye color	**Pass**	**Fail**	Pr_{pass}	Pr_{fail}
Blue	3	1	3/4	1/5
Brown	1	2	1/4	2/5
Green	0	2	0/4	2/5

Result	**Number**	**Pr**
Pass	4	4/9
Fail	5	5/9

If we now would ask about (Medium, Fat, Green), we would come to following result, when trying to calculate Pass value:
$P = 0/4 * 1/4 * 0/4 * 4/9 = 0$
That would mean that having green eyes or being medium height does not give you any chance for passing the exam, according to data. But this is only because we do not have entries with such values that passed the exam. This is called zero frequency problem. One way to fix this problem is to add fixed value to all the values. This comes from a technique called **additive smoothing** or **Laplace smoothing**. In Naive Bayes commonly value 1 is added, as simplest approach. *Although it is worth to notice that calculation of value that should be added is another problem, described in technique mentioned.*
So the best way is to rewrite our frequency tables, that we can see how our total values change as well:

Heigth	**Pass**	**Fail**	Pr_{pass}	Pr_{fail}
Short	1+1	1+1	2/7	2/8
Medium	0+1	3+1	1/7	4/8
Tall	3+1	1+1	4/7	2/8
Total	**4+3**	**5+3**	**7/7**	**8/8**

Weigth	**Pass**	**Fail**	Pr_{pass}	Pr_{fail}
Skinny	2+1	1+1	3/7	2/8
Medium	1+1	1+1	2/7	2/8
Fat	1+1	3+1	2/7	4/8
Total	**4+3**	**5+3**	**7/7**	**8/8**

Eye color	**Pass**	**Fail**	Pr_{pass}	Pr_{fail}
Blue	3+1	1+1	4/7	2/8
Brown	1+1	2+1	2/7	3/8
Green	0+1	2+1	1/7	3/8
Total	**4+3**	**5+3**	**7/7**	**8/8**

Overall we have added 3 artifical entries for Pass and 3 for Fail. Each table has also now row with Total added, where we can observe this as well as fact, that probabilities still sum up to 100%. Table with results will change as follows:

Result	**Number**	**Pr**
Pass	4+3	4+3/9+6 = 7/15
Fail	5+3	5+3/9+6 = 8/15

With such approach, we are sure that we do not have any zero probabilities and can safely perform our calculations. Just as an exercise, let us compare how probabilities have changed after this addition. Tables below show values in percents for factor Heigth and for final result.

Heigth	$P(Pass)_{bef}$	$P(Fail)_{bef}$	$P(Pass)_{aft}$	$P(Fail)_{aft}$
Short	25%	20%	28%	25%
Medium	0%	60%	14%	50%
Tall	75%	20%	58%	25%

Result	P_{bef}	P_{aft}
Pass	44%	46%
Fail	56%	54%

We can notice that probabilities for Height factor are now indeed *smoother*. More interesting, probabilities for final results did not change much even though we added 6 entries to only 9 existing ones.