# Data Science Cheat Sheet

## Probability

P(A U B) = P(A) + P(B) [assumes mutual exclusivity]

P(A U B) = P(A) + P(B) – P(A intersect B) [if A and B are non-exclusive]

MULTIPLICATION RULE

P(A ∩ B) = P(A) \* P(B) [if the events are independent]

P(A ∩ B) = P(A) \* P(B | A) [if events are dependent]

P(A ∩ B) = P(B) \* P(A | B) [if events are dependent]

A screenshot of a cell phone

Description automatically generated

A picture containing knife

Description automatically generated



A screenshot of a cell phone

Description automatically generated

A picture containing table

Description automatically generated

A screenshot of a cell phone

Description automatically generated

LAW OF TOTAL PROBABILITY

A screenshot of a cell phone

Description automatically generated

A screenshot of a cell phone

Description automatically generated

A close up of a logo

Description automatically generated

* Additive smoothing – adding a smoother parameter *α* to avoid 0 probability results for unseen events

A screenshot of a cell phone

Description automatically generated

As a side note, when *α*=1, the additive smoothing technique is most commonly known as **Laplace smoothing** (or **add-one smoothing**). However, it is also possible to use *α*<1, in which case the technique is called **Lidstone smoothing**.

Depending on the math and the assumptions used, the Naive Bayes algorithm has a few variations. The three most popular Naive Bayes algorithms are:

* Multinomial Naive Bayes
* Gaussian Naive Bayes
* Bernoulli Naive Bayes

In this mission, we learned the **multinomial Naive Bayes** version of the algorithm. Explaining the mathematical differences between the various versions is out of the scope of this course, but it's important to keep in mind that all the Naive Bayes algorithms build on the (naive) conditional independence assumption we learned about earlier in this mission.

To summarize everything we've done so far, these are the two equations we can use for our spam filtering problem moving forward:

*P*(*Spam*|*w*1,*w*2,...,*wn*)∝*P*(*Spam*)⋅∏*i*=1*nP*(*wi*|*Spam*)

*P*(*SpamC*|*w*1,*w*2,...,*wn*)∝*P*(*SpamC*)⋅∏*i*=1*nP*(*wi*|*SpamC*)

To calculate P(wi|Spam) and P(wi|SpamC), we need to use the additive smoothing technique:

*P*(*wi*|*Spam*)=*Nwi*|*Spam*+*αNSpam*+*α*⋅*NVocabulary*

*P*(*wi*|*SpamC*)=*Nwi*|*SpamC*+*αNSpamC*+*α*⋅*NVocabulary*

Let's also summarize what the terms in the equations above mean:

*Nwi*|*Spam*=the number of times the word *wi* occurs in spam messages*Nwi*|*SpamC*=the number of times the word *wi* occurs in non-spam messages*NSpam*=total number of words in spam messages*NSpamC*=total number of words in non-spam messages*NVocabulary*=total number of words in the vocabulary*α*=1    (*α* is a smoothing parameter)

It's worth emphasizing that:

* **NSpam** is equal to the number of words in all the spam messages — it's *not* equal to the number of spam messages, and it's not equal to the total number of *unique* words in spam messages.
* **NSpamC** is equal to the number of words in all the non-spam messages — it's *not* equal to the number of non-spam messages, and it's not equal to the total number of *unique* words in non-spam messages.