**Naïve Bayes Classifiers:**

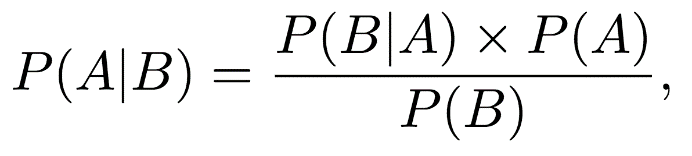
**Case Study:**

We want to classify if something belongs to the Entertainment, Computer Science, or Zoology class. We know that most of the data is Entertainment. We then get a new data input e.g. “Python”, this could be either Zoology (the snake), Computer Science (the programming language), or Entertainment (Monty Python) …

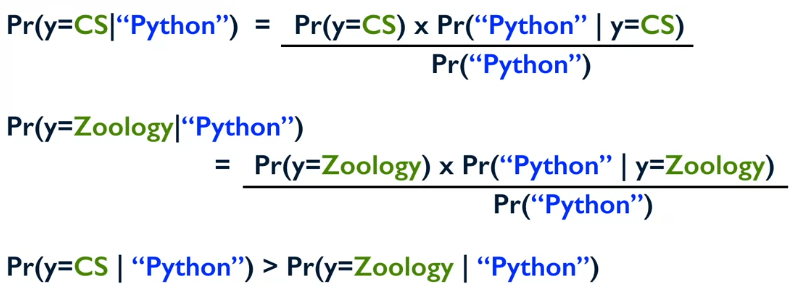
Evaluating the word on its own it is likely to be classified as Zoology, however, if the data was “Python download” then it would be Computer Science.

Naïve Bayes works by updating its likelihood of data being in one class or the other by this new data. At the beginning the most likely class is the Entertainment class, but when we input “python” as a new entry it then become unlikely that this is entertainment, so it predicts Zoology.

The model starts with having a **prior probability** which is due to the class balance and the training data. Then with new data inputs e.g. “python” the probability changes (**posterior probability**) and it become more likely to be in the Zoology class. This is part of **Bayes’ Rule**



The equation above says that the probability of A given B (where A is the classes and B is data) is equal to the **prior probability** Pr(A), which is based on the class balance, times the probability of the data given the classes (**likelihood**) divided by the probability of the data. <https://towardsdatascience.com/probability-concepts-explained-bayesian-inference-for-parameter-estimation-90e8930e5348> for an example.



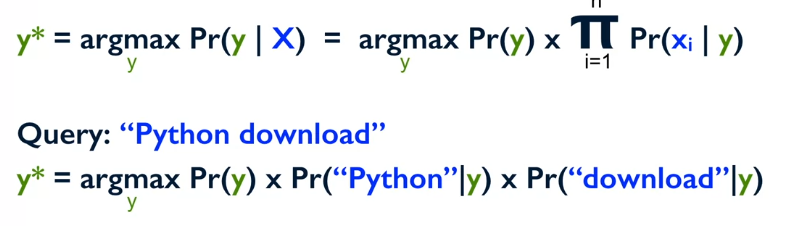
It computes the probability of the class compared to the data and then returns the most likely class! However, because we only care about the largest probability of the class given the data entry, we actually don’t care about the probability of the data entry occurring (as this is constant) so the equation becomes:



We want to return the class value (y) that has the highest value of the prior probability times the likelihood of the data (X) occurring in that class.

**Naïve Bayes Assumption: The presence of a particular feature in a class is unrelated to the presence of any other feature.** E.g. “White House”, given that the word “White” has a capital W then its likely that “House” will follow, but Naïve Bayes does not make this connection so it is “naïve”.

**Example:**



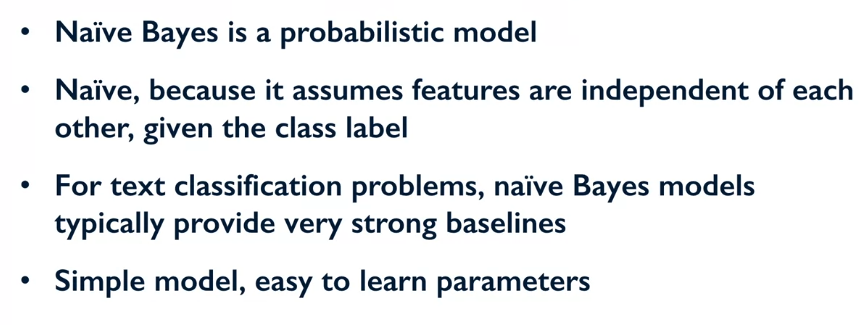
This is then done for all y in Y and the largest value of y is returned as the class.

**Naïve Bayes: Smoothing:**

What happens when the probability of x in y is 0? For example, what if in the training data your model has never seen the word “Python”, and now its trying to predict which class is should go into?

To counteract this problem, you can just add a **dummy count** to all the instances of all words. E.g. if python does occur in the data once then count it as occurring once, if data appears 3 times then count it as appearing 4 times. This doesn’t effect the probability significantly for large sets of data.

**Summary:**



**Naïve Bayes Variations:**

**Two Class Variants for Text:**

* Multinomial Naïve Bayes
* Bernoulli Naïve Bayes

**Multinomial Naïve Bayes:**

Here we assume that the data follows a multinomial distribution, this means that for each individual data entry its features are independent of each other, and that the count of the features is now important. Each feature would become a count e.g. simple count or TF-IDF weighting.

A **bag of words** models is when you find out all the words used in the dataset. If your features were if the word was present or not (0 or 1) then this would be a Bernoulli distribution. You could also look at counting the number of times words appear in a corpus of words, on top of this if you think that the infrequent words are more important you can use a term **frequency inverse document frequency weighting (TF-IDF)**. If you wanted to give importance to rare words you could add in this weighting.

**Bernoulli Naïve Bayes:**

The assumption here is that the data follows a multivariate Bernoulli distribution, so each feature is a binary feature (word present or not present). Here the frequency of the words in the corpus doesn’t matter.