‘**Naive Bayes**‘, which can be **extremely, fast** relative to other classification algorithms

It is a classification technique based on [Bayes’ Theorem](https://en.wikipedia.org/wiki/Bayes%27_theorem) with an assumption of **independence among predictors.**

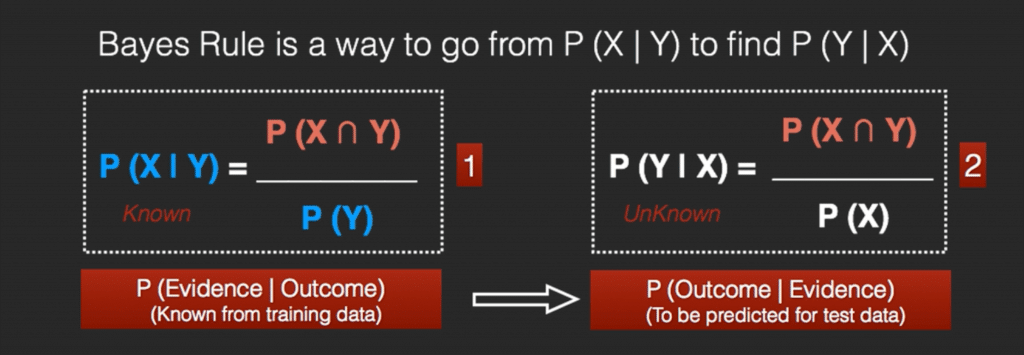
Naive Bayes model is **easy to build** and **particularly useful for very large data sets**. Along with simplicity, Naive Bayes is known to **outperform even highly sophisticated classification methods**.

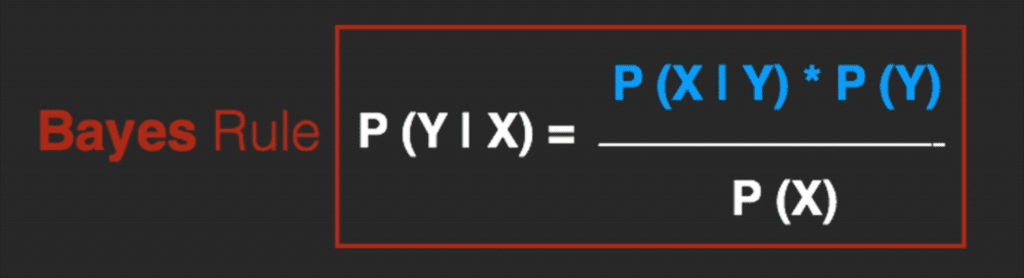
## The Bayes Rule

The Bayes Rule is a way of going from P(X|Y), known from the training dataset, to find P(Y|X).

For observations in test or scoring data, the X would be known while Y is unknown. And for each row of the test dataset, you want to compute the probability of Y given the X has already happened.

What happens if Y has more than 2 categories? We compute the probability of each class of Y and let the highest win.

[](https://www.machinelearningplus.com/wp-content/uploads/2018/11/01_bayes_rule_derive_new.png)

[](https://www.machinelearningplus.com/wp-content/uploads/2018/11/02_bayes_rule_new.png)

## 

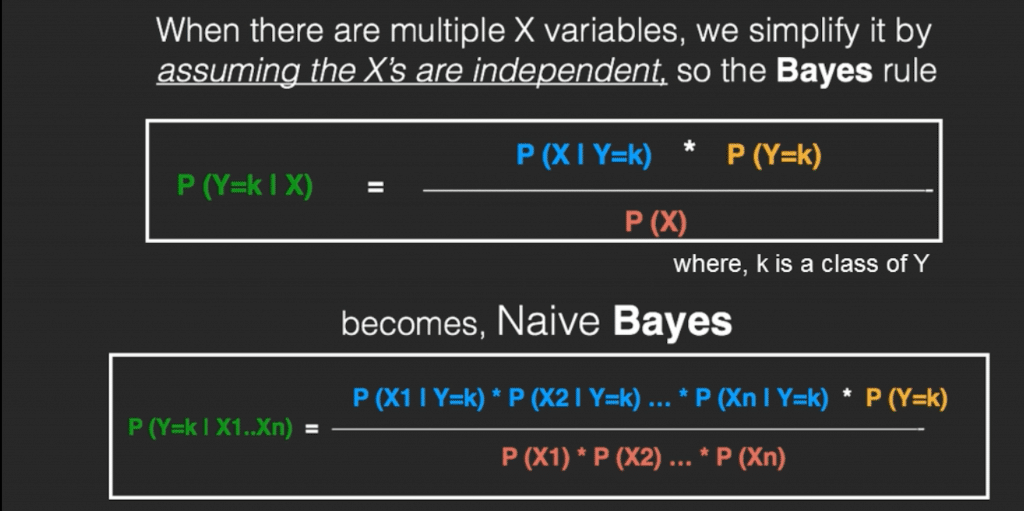
## 

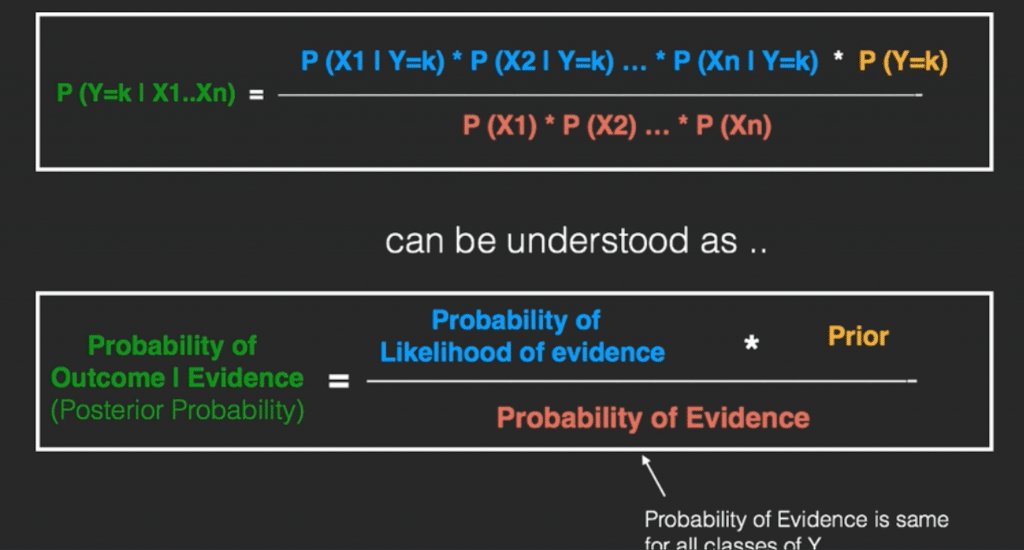
## The Naive Bayes

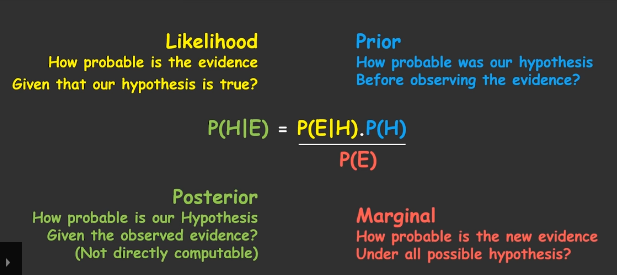
The Bayes Rule provides the formula for the probability of Y given X. But, in real-world problems, you typically have multiple X variables.

When the features are independent, we can extend the Bayes Rule to what is called Naive Bayes.

It is called ‘Naive’ because of the naive assumption that the X’s are independent of each other. Regardless of its name, it’s a powerful formula.

[](https://www.machinelearningplus.com/wp-content/uploads/2018/11/03_bayes_rule_naive_bayes_new.png)

[](https://www.machinelearningplus.com/wp-content/uploads/2018/11/04_naive_bayes_interpretation_new.png)



In technical jargon, the left-hand-side (LHS) of the equation is understood as the posterior probability or simply the posterior

The RHS has 2 terms in the numerator.

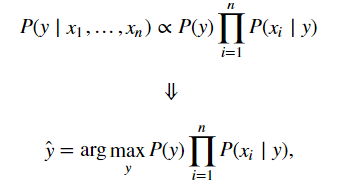
The first term is called the **‘Likelihood of Evidence’**. It is nothing but the conditional probability of each X’s given Y is of particular class ‘c’.

Since all the X’s are assumed to be independent of each other, you can just multiply the ‘likelihoods’ of all the X’s and called it the ‘Probability of likelihood of evidence’. This is known from the training dataset by filtering records where Y=c.

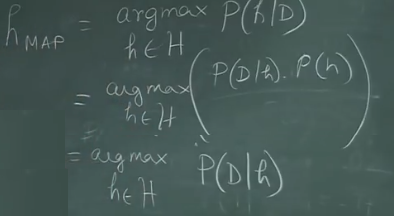
The second term is called the prior which is the overall probability of Y=c, where c is a class of Y. In simpler terms, Prior = count(Y=c) / n\_Records.

**Classifier combines this model with a decision rule; this decision rule will decide which hypothesis is most probable. Picking the hypothesis that is most probable is known as the maximum a posteriori or MAP decision rule**. The corresponding classifier, a Bayes classifier, is the function that assigns a class label to y as follows:

Since P(x1, …, xn) is constant given the input, we can use the following classification rule:



We can use Maximum A Posteriori (MAP) estimation to estimate P(y) and P(xi | y); the former is then the relative frequency of class y in the training set



**Naive Bayes Example by Hand**

Say you have 1000 fruits which could be either ‘banana’, ‘orange’ or ‘other’. These are the 3 possible classes of the Y variable.

We have data for the following X variables, all of which are binary (1 or 0).

* Long
* Sweet
* Yellow

The first few rows of the training dataset look like this:

| **Fruit** | **Long (x1)** | **Sweet (x2)** | **Yellow (x3)** |
| --- | --- | --- | --- |
| Orange | 0 | 1 | 0 |
| Banana | 1 | 0 | 1 |
| Banana | 1 | 1 | 1 |
| Other | 1 | 1 | 0 |
| .. | .. | .. | .. |

For the sake of computing the probabilities, let’s aggregate the training data to form a counts table like this.

[](https://www.machinelearningplus.com/wp-content/uploads/2018/11/05_Naive_bayes_example_new.png)

So the objective of the classifier is to predict if a given fruit is a ‘Banana’ or ‘Orange’ or ‘Other’ when only the 3 features (long, sweet and yellow) are known.

Let’s say you are given a fruit that is: Long, Sweet and Yellow, can you predict what fruit it is?

This is the same of predicting the Y when only the X variables in testing data are known. Let’s solve it by hand using Naive Bayes.

The idea is to compute the 3 probabilities, that is the probability of the fruit being a banana, orange or other. Whichever fruit type gets the highest probability wins.

All the information to calculate these probabilities is present in the above tabulation.

**Step 1: Compute the ‘Prior’ probabilities for each of the class of fruits.**

That is, the proportion of each fruit class out of all the fruits from the population. You can provide the ‘Priors’ from prior information about the population. Otherwise, it can be computed from the training data.

For this case, let’s compute from the training data. Out of 1000 records in training data, you have 500 Bananas, 300 Oranges and 200 Others. So the respective priors are 0.5, 0.3 and 0.2.

P(Y=Banana) = 500 / 1000 = 0.50

P(Y=Orange) = 300 / 1000 = 0.30

P(Y=Other) = 200 / 1000 = 0.20

**Step 2: Compute the probability of evidence that goes in the denominator.**

This is nothing but the product of P of Xs for all X. This is an optional step because the denominator is the same for all the classes and so will not affect the probabilities.

P(x1=Long) = 500 / 1000 = 0.50

P(x2=Sweet) = 650 / 1000 = 0.65

P(x3=Yellow) = 800 / 1000 = 0.80

**Step 3: Compute the probability of likelihood of evidences that goes in the numerator.**

It is the product of conditional probabilities of the 3 features. If you refer back to the formula, it says P(X1 |Y=k). Here X1 is ‘Long’ and k is ‘Banana’. That means the probability the fruit is ‘Long’ given that it is a Banana. In the above table, you have 500 Bananas. Out of that 400 is long. So, P(Long | Banana) = 400/500 = 0.8.

Here, I have done it for Banana alone.

**Probability of Likelihood for Banana**

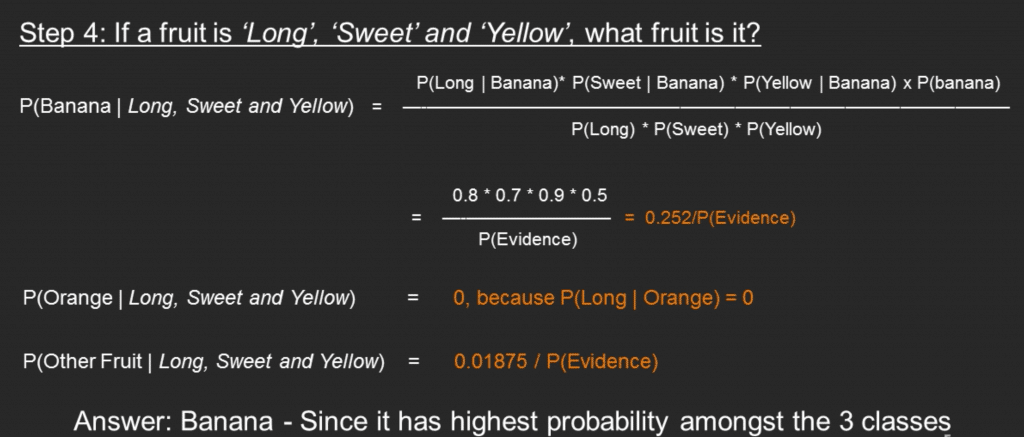
P(x1=Long | Y=Banana) = 400 / 500 = 0.80

P(x2=Sweet | Y=Banana) = 350 / 500 = 0.70

P(x3=Yellow | Y=Banana) = 450 / 500 = 0.90

So, the overall probability of Likelihood of evidence for Banana = 0.8 \* 0.7 \* 0.9 = 0.504

**Step 4: Substitute all the 3 equations into the Naive Bayes formula, to get the probability that it is a banana.**

[](https://www.machinelearningplus.com/wp-content/uploads/2018/11/06_Naive_bayes_example_answer_new.png)  
Similarly, you can compute the probabilities for ‘Orange’ and ‘Other fruit’. The denominator is the same for all 3 cases, so it’s optional to compute.

Clearly, Banana gets the highest probability, so that will be our predicted class.

## What is Laplace Correction?

The value of P(Orange | Long, Sweet and Yellow) was zero in the above example, because, P(Long | Orange) was zero. That is, there were no ‘Long’ oranges in the training data.

It makes sense, but when you have a model with many features, the entire probability will become zero because one of the feature’s value was zero. To avoid this, we increase the count of the variable with zero to a small value (usually 1) in the numerator, so that the overall probability doesn’t become zero.

This correction is called ‘Laplace Correction’. Most Naive Bayes model implementations accept this or an equivalent form of correction as a parameter.

### Types of Naive Bayes Classifier:

#### Multinomial Naive Bayes:

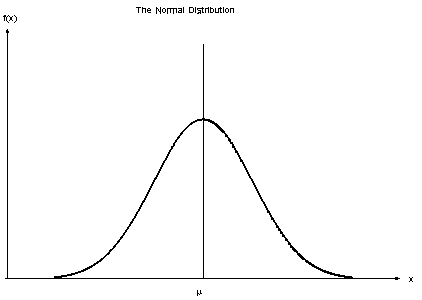
This is mostly used for document classification problem, i.e whether a document belongs to the category of sports, politics, technology etc. The features/predictors used by the classifier are the frequency of the words present in the document.

#### Bernoulli Naive Bayes:

This is similar to the multinomial naive bayes but the predictors are boolean variables. The parameters that we use to predict the class variable take up only values yes or no, for example if a word occurs in the text or not.

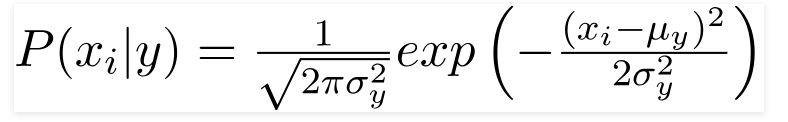
#### Gaussian Naive Bayes:

When the predictors take up a continuous value and are not discrete, we assume that these values are sampled from a gaussian distribution.

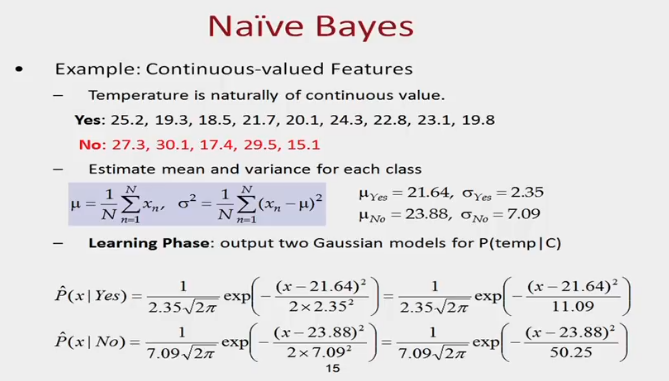


Gaussian Distribution(Normal Distribution)

Since the wa



|  |  |  |
| --- | --- | --- |
| Numerical variables need to be transformed to their categorical counterparts ([binning](https://www.saedsayad.com/binning.htm)) before constructing their frequency tables. The other option we have is using the distribution of the numerical variable to have a good guess of the frequency. For example, one common practice is to assume normal distributions for numerical variables. |  |  |
|  |  |  |
| The probability density function for the normal distribution is defined by two parameters (mean and standard deviation). |  |  |
| https://www.saedsayad.com/images/Bayes_NormDist.png |  |  |
|  |  |  |
| *Example*: |  |  |
| |  |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | |  |  | **Humidity** | | | | | | | |  | *Mean* | *StDev* | | **Play Golf** | yes | 86 | 96 | 80 | 65 | 70 | 80 | 70 | 90 | 75 | 79.1 | 10.2 | | no | 85 | 90 | 70 | 95 | 91 |  |  |  |  | 86.2 | 9.7 | |  |  |
|  |  |  |
| https://www.saedsayad.com/images/Bayes_NormDist_1.png |  |  |
|  |  |  |



**Tips to improve the model**

1. Try transforming the variables using transformations like BoxCox or YeoJohnson to make the features near Normal.
2. Try applying Laplace correction to handle records with zeros values in X variables.
3. Check for correlated features and try removing the highly correlated ones. Naive Bayes is based on the assumption that the features are independent.
4. Feature engineering. Combining features (a product) to form new ones that makes intuitive sense might help.
5. Try providing more realistic prior probabilities to the algorithm based on knowledge from business, instead of letting the algo calculate the priors based on the training sample.

For this case, ensemble methods like bagging, boosting will help a lot by reducing the variance.

==============================================================

Bayes’ Theorem provides a way that we can calculate the probability of a hypothesis given our prior knowledge.

Bayes’ Theorem is stated as:

P(h|d) = (P(d|h) \* P(h)) / P(d)

Where

* **P(h|d)** is the probability of hypothesis h given the data d. This is called the posterior probability.T**he conditional probability that event h occurs , given that d has occurred. This is also known as the posterior probability.**
* **P(d|h)** is the **probability** of data d given that the hypothesis h was true. **The conditional probability that event d occurs , given that h has occurred. Called PRIOR**
* **P(h)** is the probability of hypothesis h being true (regardless of the data). This is called the prior probability of h.
* **P(d)** is the probability of the data (regardless of the hypothesis).
* **P(A|B)** : conditional probability of response variable belonging to a particular value, given the input attributes. **This is also known as the posterior probability.**
* **P(A)** : **The prior probability of the response variable.**
* **P(B) : The probability of training data or the evidence.**
* **P(B|A) : This is known as the likelihood of the training data.**

**What are the Pros and Cons of Naive Bayes?**

***Pros:***

* It is easy and fast to predict class of test data set. It also perform well in multi class prediction
* When assumption of independence holds, a Naive Bayes classifier performs better compare to other models like logistic regression and you need less training data.
* It perform well in case of categorical input variables compared to numerical variable(s). For numerical variable, normal distribution is assumed (bell curve, which is a strong assumption).

***Cons:***

* If categorical variable has a category (in test data set), which was not observed in training data set, then model will assign a 0 (zero) probability and will be unable to make a prediction. This is often known as “Zero Frequency”. To solve this, we can use the smoothing technique. One of the simplest smoothing techniques is called Laplace estimation.
* On the other side naive Bayes is also known as a bad estimator, so the probability outputs from predict\_proba are not to be taken too seriously.
* Another limitation of Naive Bayes is the assumption of independent predictors. In real life, it is almost impossible that we get a set of predictors which are completely independent.

**Applications of Naive Bayes Algorithms**

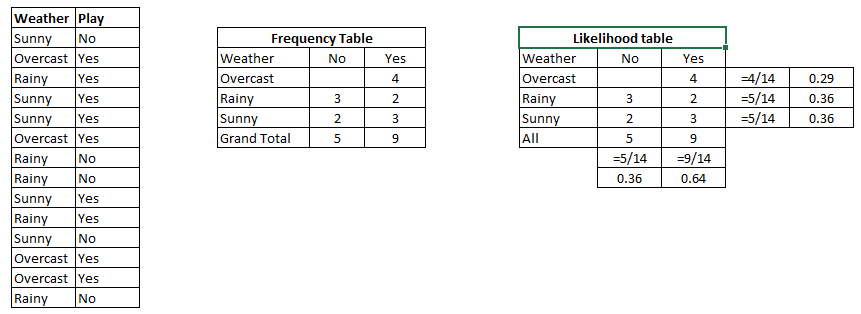
* **Real time Prediction:**Naive Bayes is an eager learning classifier and it is sure fast. Thus, it could be used for making predictions in real time.
* **Multi class Prediction:**This algorithm is also well known for multi class prediction feature. Here we can predict the probability of multiple classes of target variable.
* **Text classification/ Spam Filtering/ Sentiment Analysis:** Naive Bayes classifiers mostly used in text classification (due to better result in multi class problems and independence rule) have higher success rate as compared to other algorithms. As a result, it is widely used in Spam filtering (identify spam e-mail) and Sentiment Analysis (in social media analysis, to identify positive and negative customer sentiments)
* **Recommendation System:**Naive Bayes Classifier and [Collaborative Filtering](https://en.wikipedia.org/wiki/Collaborative_filtering) together builds a Recommendation System that uses machine learning and data mining techniques to filter unseen information and predict whether a user would like a given resource or not
* **Medical diagnosis:** Well suited for disease diagnosis
* **Weather prediction:**

## How Naive Bayes algorithm works?

Let’s understand it using an example. Below I have a training data set of weather and corresponding target variable ‘Play’ (suggesting possibilities of playing). Now, we need to classify whether players will play or not based on weather condition. Let’s follow the below steps to perform it.

Step 1: Convert the data set into a frequency table

Step 2: Create Likelihood table by finding the probabilities like Overcast probability = 0.29 and probability of playing is 0.64.

[](https://www.analyticsvidhya.com/wp-content/uploads/2015/08/Bayes_41.png)

Step 3: Now, use Naive Bayesian equation to calculate the posterior probability for each class. The class with the highest posterior probability is the outcome of prediction.

**Problem:** Players will play if weather is sunny. Is this statement is correct?

We can solve it using above discussed method of posterior probability.

P(Yes | Sunny) = P( Sunny | Yes) \* P(Yes) / P (Sunny)

Here we have P (Sunny |Yes) = 3/9 = 0.33, P(Sunny) = 5/14 = 0.36, P( Yes)= 9/14 = 0.64

Now, P (Yes | Sunny) = 0.33 \* 0.64 / 0.36 = 0.60, which has higher probability.

Naive Bayes uses a similar method to predict the probability of different class based on various attributes. This algorithm is mostly used in text classification and with problems having multiple classes

|  |  |  |
| --- | --- | --- |
| *Example 1*: |  |  |
| We use the same simple Weather dataset here. |  |  |
| https://www.saedsayad.com/images/naive_bayes_data.png |  |  |
| The posterior probability can be calculated by first, constructing a frequency table for each attribute  against the target. Then, transforming the frequency tables to likelihood tables and finally use  the Naive Bayesian equation to calculate the posterior probability for each class. The class with the  highest posterior probability is the outcome of prediction. |  |  |
| https://www.saedsayad.com/images/Bayes_3.png |  |  |
| The likelihood tables for all four predictors. |  |  |
| https://www.saedsayad.com/images/naive_bayes_likelihood.png |  |  |
|  |  |  |
| *Example 2*: |  |  |
| In this example we have 4 inputs (predictors). The final **posterior** probabilities can be standardized between 0 and 1. |  |  |
| https://www.saedsayad.com/images/naive_bayes_example_2.png |  |  |
|  |  |  |
| **The zero-frequency problem** |  |  |
| Add 1 to the count for every attribute value-class combination (*Laplace estimator*) when an attribute value (*Outlook=Overcast*) doesn’t occur with every class value (*Play Golf=no*). |  |  |
|  |  |  |
| **Spam Classification Python Example:** |  |  |