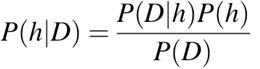
**What is Naive Bayes Classifier?**

Naive Bayes is a statistical classification technique based on Bayes Theorem. It is one of the simplest supervised learning algorithms. Naive Bayes classifier is the fast, accurate and reliable algorithm. Naive Bayes classifiers have high accuracy and speed on large datasets.

Naive Bayes classifier assumes that the effect of a particular feature in a class is independent of other features. For example, a loan applicant is desirable or not depending on his/her income, previous loan and transaction history, age, and location. Even if these features are interdependent, these features are still considered independently. This assumption simplifies computation, and that's why it is considered as naive. This assumption is called class conditional independence.



* P(h): the probability of hypothesis h being true (regardless of the data). This is known as the prior probability of h.
* P(D): the probability of the data (regardless of the hypothesis). This is known as the prior probability.
* P(h|D): the probability of hypothesis h given the data D. This is known as posterior probability.
* P(D|h): the probability of data d given that the hypothesis h was true. This is known as posterior probability.

**How Naive Bayes Classifier Works?**

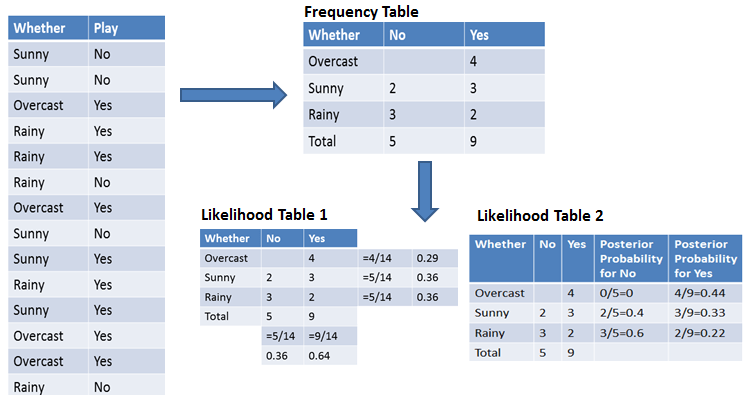
Let’s understand the working of Naive Bayes through an example. Given an example of weather conditions and playing sports. You need to calculate the probability of playing sports. Now, you need to classify whether players will play or not, based on the weather condition.

**First Approach (In case of a single feature)**

Naive Bayes classifier calculates the probability of an event in the following steps:

* **Step 1**: Calculate the prior probability for given class labels
* **Step 2**: Find Likelihood probability with each attribute for each class
* **Step 3**: Put these value in Bayes Formula and calculate posterior probability.
* **Step 4**: See which class has a higher probability, given the input belongs to the higher probability class.

For simplifying prior and posterior probability calculation, you can use the two tables frequency and likelihood tables. Both of these tables will help you to calculate the prior and posterior probability. The Frequency table contains the occurrence of labels for all features. There are two likelihood tables. Likelihood Table 1 is showing prior probabilities of labels and Likelihood Table 2 is showing the posterior probability.



Now suppose you want to calculate the probability of playing when the weather is overcast.

**Probability of playing:**

*P(Yes | Overcast) = P(Overcast | Yes)*P(Yes) / P (Overcast) .....................(1)

1. Calculate Prior Probabilities:

P(Overcast) = 4/14 = 0.29

P(Yes)= 9/14 = 0.64

1. Calculate Posterior Probabilities:

P(Overcast |Yes) = 4/9 = 0.44

1. Put Prior and Posterior probabilities in equation (1)

P (Yes | Overcast) = 0.44 \* 0.64 / 0.29 = 0.98(Higher)

Similarly, you can calculate the probability of not playing:

**Probability of not playing:**

*P(No | Overcast) = P(Overcast | No)*P(No) / P (Overcast) .....................(2)

1. Calculate Prior Probabilities:

P(Overcast) = 4/14 = 0.29

P(No)= 5/14 = 0.36

1. Calculate Posterior Probabilities:

P(Overcast |No) = 0/9 = 0

1. Put Prior and Posterior probabilities in equation (2)

P (No | Overcast) = 0 \* 0.36 / 0.29 = 0

*The probability of a 'Yes' class is higher. So you can determine here if the weather is overcast than players will play the sport.*

FOR REFERENCE, visit:

https://saskeli.github.io/data-analysis-with-python-summer-2019/bayes.html