Naïve Bayes is a simple technique for building classifiers: models that assign class labels to problem instances. The basic idea of Bayes rule is that the outcome of a hypothesis can be predicted on the basis of some evidence (*E*) that can be observed.

From Bayes rule, we observed that

1. A prior probability of hypothesis *h* or *P*(*h*): This is the probability of an event or hypothesis before the evidence is observed.
2. A posterior probability of *h* or *P*(*h*|*D*): This is the probability of an event after the evidence is observed within the population *D*.

$$\textbf{Posterior probability} = \frac{\textbf{(Prior probability} \times \textbf{Conditional Probability)}}{\textbf{Evidence}}$$

Posterior Probability is of the format 'What is the probability that a particular object belongs to class i given its observed feature values?'

For example, a person has height and weight of 182 cm and 68 kg, respectively. What is the probability that this person belongs to the class 'basketball player'? This can be predicted using the Naïve Bayes classifier. This is known as probabilistic classifications.

In machine learning, a probabilistic classifier is a classifier that can be foreseen, given a perception or information (input), a likelihood calculation over a set of classes, instead of just yielding (outputting) the most likely class that the perception (observation) should belong to. Parameter estimation for Naïve Bayes models uses the method of ML.

Bayes' theorem is used when new information can be used to revise previously determined probabilities. Depending on the particular nature of the probability model, Naïve Bayes classifiers can be trained very professionally in a supervised learning setting.

Let us see the basis of deriving the principles of Naïve Bayes classifiers. We take a learning task where each instance *x* has some attributes and the target function ($f(x)$) can take any value from the finite set of classification values C. We also have a set of training examples for target function, and the set of attributes $\{a_1, a_2, \ldots, a_n\}$ for the new instance are known to us. Our task is to predict the classification of the new instance.

According to the approach in Bayes' theorem, the classification of the new instance is performed by assigning the most probable target classification C_{MAP} on the basis of the attribute values of the new instance $\{a_1, a_2, \ldots, a_n\}$. So,

$$C_{\mathrm{MAP}} = \underset{c_i \in C}{\operatorname{argmax}} \sum_{h_i \in H} P\left(c_i | a_1, a_2, \ldots, a_n\right)$$

which can be rewritten using Bayes' theorem as

$$C_{\mathrm{MAP}} = \underset{c_i \in C}{\operatorname{argmax}} \sum_{h_i \in H} \frac{P(a_1, a_2, \ldots, a_n | c_i) P\left(c_i\right)}{P\left(a_1, \ldots a_n\right)}$$

As combined probability of the attributes defining the new instance fully is always 1

$$C_{\text{MAP}} = \underset{c_i \in C}{\text{argmax}} \sum_{h_i \in H} P(a_1,\ a_2, \ldots, a_n | c_i) P(c_i) \tag{6.8}$$

So, to get the most probable classifier, we have to evaluate the two terms $P(a_1, a_2, \text{c}, a_n|c_i)$ and $P(ci)$. In a practical scenario, it is possible to calculate $P(ci)$ by calculating the frequency of each target value c_i in the training data set. But the $P(a_1, a_2, \text{c}, a_n|c_i)$ cannot be estimated easily and needs a very high effort of calculation. The reason is that the number of these terms is equal to the product of number of possible instances and the number of possible target values, and thus, each instance in the instance space needs to be visited many times to arrive at the estimate of the occurrence. Thus, the Naïve Bayes classifier makes a simple assumption that the attribute values are conditionally independent of each other for the target value. So, applying this simplification, we can now say that for a target value of an instance, the probability of observing the combination $a_1,a_2,\ldots, a_n$ is the product of probabilities of individual attributes $P(a_i|c_j)$.

$$P(a_1,\ a_2, ..., a_n|c_j) = \prod_i P(a_i|c_j)$$

Then, from equation 6.7, we get the approach for the Naïve Bayes classifier as

$$\text{C}_{\text{NB}} = \underset{c_i \in C}{\text{argmax}} \sum_{h_i \in H} P(c_i) \prod_i P(a_i|c_j) \tag{6.9}$$

Here, we will be able to compute $P(a_i|c_j)$ as we have to calculate this only for the number of distinct attributes values (a_i) times the number of distinct target values (c_j), which is much smaller set than the product of both the sets. The most important reason for the popularity of the Naïve Bayes classifier approach is that it is not required to search the whole hypothesis space for this algorithm, but rather we can arrive at the target classifier by simply counting the frequencies of various data combinations within the training example.

To summarize, a Naïve Bayes classifier is a primary probabilistic classifier based on a view of applying Bayes' theorem (from Bayesian inference with strong naive) independence assumptions. The prior probabilities in Bayes' theorem that are changed with the help of newly available information are classified as posterior probabilities.

A key benefit of the naive Bayes classifier is that it requires only a little bit of training information (data) to gauge the parameters (mean and differences of the variables) essential for the classification (arrangement). In the Naïve Bayes classifier, independent variables are always assumed, and only the changes (variances) of the factors/variables for each class should be determined and not the whole covariance matrix. Because of the rather naïve assumption that all features of the dataset are equally important and independent, this is called Naïve Bayes classifier.

Naïve Bayes classifiers are direct linear classifiers that are known for being the straightforward, yet extremely proficient result. The modified version of Naïve Bayes classifier originates from the assumption that information collection (data set) is commonly autonomous (mutually independent). In most of the practical scenarios, the 'independence' assumption is regularly violated. However, Naïve Bayes classifiers still tend to perform exceptionally well.

Some of the key strengths and weaknesses of Naïve Bayes classifiers are described in Table 6.1.

Table 6.1 *Strengths and Weaknesses of Bayes Classifiers*

Strengths	Weakness
Simple and fast in calculation but yet effective in result	The basis assumption of equal importance and independence often does not hold true
In situations where there are noisy and missing data, it performs well	If the target dataset contains large numbers of numeric features, then the reliability of the outcome becomes limited
Works equally well when smaller number of data is present for training as well as very large number of training data is available	Though the predicted classes have a high reliability, estimated probabilities have relatively lower reliability
Easy and straightforward way to obtain the estimated probability of a prediction	

Example. Let us assume that we want to predict the outcome of a football world cup match on the basis of the past performance data of the playing teams. We have training data available (refer Fig. 6.3) for actual match outcome, while four parameters are considered – Weather Condition (Rainy, Overcast, or Sunny), how many matches won were by this team out of the last three matches (one match, two matches, or three matches), Humidity Condition (High or Normal), and whether they won the toss (True or False). Using Naïve Bayesian, you need to classify the conditions when this team wins and then predict the probability of this team winning a particular match when Weather Conditions = Rainy, they won two of the last three matches, Humidity = Normal and they won the toss in the particular match.

Weather Condition	Wins in last 3 matches	Humidity	Win toss	Won match?
Rainy	3 wins	High	FALSE	No
Rainy	3 wins	High	TRUE	No
OverCast	3 wins	High	FALSE	Yes
Sunny	2 wins	High	FALSE	Yes
Sunny	1 win	Normal	FALSE	Yes
Sunny	1 win	Normal	TRUE	No
OverCast	1 win	Normal	TRUE	Yes
Rainy	2 wins	High	FALSE	No
Rainy	1 win	Normal	FALSE	Yes
Sunny	2 wins	Normal	FALSE	Yes
Rainy	2 wins	Normal	TRUE	Yes
OverCast	2 wins	High	TRUE	Yes
OverCast	3 wins	Normal	FALSE	Yes
Sunny	2 wins	High	TRUE	No

FIG. 6.3 Training data for the Naïve Bayesian method

6.4.4.1 Naïve Bayes classifier steps

Step 1: First construct a frequency table. A frequency table is drawn for each attribute against the target outcome. For example, in Figure 6.3, the various attributes are (1) Weather Condition, (2) How many matches won by this team in last three matches, (3) Humidity Condition, and (4) whether they won the toss and the target outcome is will they win the match or not?

Step 2: Identify the cumulative probability for ‘Won match = Yes’ and the probability for ‘Won match = No’ on the basis of all the attributes. Otherwise, simply multiply probabilities of all favourable conditions to derive ‘YES’ condition. Multiply probabilities of all non-favourable conditions to derive ‘No’ condition.

Step 3: Calculate probability through normalization by applying the below formula

$$P(\text{Yes}) = \frac{P(\text{Yes})}{P(\text{Yes}) + P(\text{No})}$$

$$P(\text{No}) = \frac{P(\text{No})}{P(\text{Yes}) + P(\text{No})}$$

P(Yes) will give the overall probability of favourable condition in the given scenario.

P(No) will give the overall probability of non-favourable condition in the given scenario.

Solving the above problem with Naive Bayes

Step 1: Construct a frequency table. The posterior probability can be easily derived by constructing a frequency table for each attribute against the target. For example, frequency of Weather Condition variable with values 'Sunny' when the target value Won match is 'Yes', is, 3/(3+4+2) = 3/9.

Figure 6.4 shows the frequency table thus constructed.

Step 2:

To predict whether the team will win for given weather conditions (a_1) = Rainy, Wins in last three matches (a_2) = 2 wins, Humidity (a_3) = Normal and Win toss (a_4) = True, we need to choose 'Yes' from the above table for the given conditions.

From Bayes' theorem, we get

$$P(\text{Win match} \mid a_1 \cap a_2 \cap a_3 \cap a_4) = \frac{P(a_1 \cap a_2 \cap a_3 \cap a_4 \mid \text{Win match})\, P(\text{Win match})}{P(a_1 \cap a_2 \cap a_3 \cap a_4)}$$

This equation becomes much easier to resolve if we recall that Naïve Bayes classifier assumes independence among events. This is specifically true for class-conditional independence, which means that the events are independent so long as they are conditioned on the same class value. Also, we know that if the events are independent, then the probability rule says, $P(A \cap B) = P(A)$ $P(B)$, which helps in simplifying the above equation significantly as

$$P(\text{Win match}|a_1 \cap a_2 \cap a_3 \cap a_4)$$

$$= \frac{P(a_1|\text{Win match})P(a_2|\text{Win match})P(a_3|\text{Win match})P(a_4|\text{Win match})P(\text{Win match})}{P(a_1)P(a_2)\mathrm{P}(a_3)\mathrm{P}(a_4)}$$
$$= 2/9*4/9*6/9*9/14$$
$$= 0.014109347$$

This should be compared with

$$P(!\text{Win match}|a_1 \cap a_2 \cap a_3 \cap a_4)$$

$$= \frac{P(a_1|!\text{Win match})P(a_2|!\text{Win match})P(a_3|!\text{Win match})P(a_4|!\text{Win match})P(!\text{Win match})}{P(a_1)P(a_2)P(a_3)P(a_4)}$$
$$= 3/5*2/5*1/5*5/14$$
$$= 0.010285714$$

	Won Match			**Won Match**	
Weather condition	Yes	No	**Humidity**	Yes	No
Sunny	3	2	High	3	4
OverCast	4	0	Normal	6	1
Rainy	2	3			
Total	9	5	Total	9	5

	Won Match			**Won Match**	
Wins in last 3 matches	Yes	No	Win toss	Yes	No
3 wins	2	2	FALSE	6	2
1 win	4	2	TRUE	3	3
2 wins	3	1			
Total	9	5	Total	9	5

FIG. 6.4 Construct frequency table

Step 3: by normalizing the above two probabilities, we can ensure that the sum of these two probabilities is 1.

$$
\begin{aligned}
P(\textbf{Win match}) &= \frac{P(\textbf{Win match})}{P(\textbf{Win match}) + P(\textbf{!Win match})} \\
&= \frac{0.014109347}{0.014109347 + 0.010285714} \\
&= 0.578368999
\end{aligned}
$$

$$
\begin{aligned}
P(\textbf{!Win match}) &= \frac{P(\textbf{!Win match})}{P(\textbf{Win match}) + P(\textbf{!Win match})} \\
&= \frac{0.010285714}{0.014109347 + 0.010285714} \\
&= 0.421631001
\end{aligned}
$$

Conclusion: This shows that there is 58% probability that the team will win if the above conditions become true for that particular day. Thus, Naïve Bayes classifier provides a simple yet powerful way to consider the influence of multiple attributes on the target outcome and refine the uncertainty of the event on the basis of the prior knowledge because it is able to simplify the calculation through independence assumption.

6.4.5 Applications of Naïve Bayes classifier

Text classification: Naïve Bayes classifier is among the most successful known algorithms for learning to classify text documents. It classifies the document where the probability of classifying the text is more. It uses the above algorithm to check the permutation and combination of the probability of classifying a document under a particular ‘Title’. It has various applications in document categorization, language detection, and sentiment detection, which are very useful for traditional retailers, e-retailors, and other businesses on judging the sentiments of their clients on the basis of keywords in feedback forms, social media comments, etc.