Dharani J

Follow 77 Followers

# All About ML — Part 9: Naïve Bayes Algorithm

Dharani J Mar 28 · 5 min read

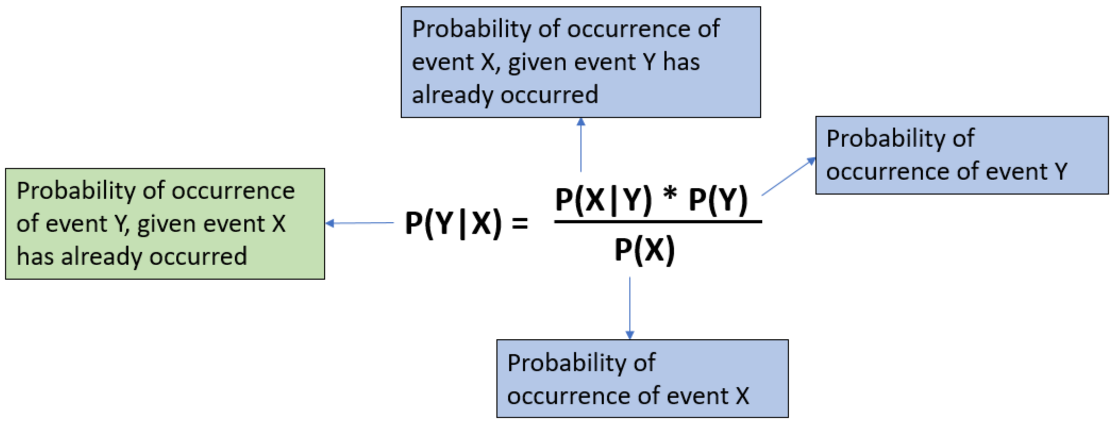
Naïve Bayes algorithm is a supervised learning algorithm. But real world data might be huge, high dimensional and it is more often that the predictor will have more than 2 classes. NB algorithm is a simple yet powerful concept which works really well on multi class variables with fast performance. NB is a combination of results obtained by using Bayes Principle on each feature to classify a target variable.

The Naïve Bayes algorithm is comprised of two words Naïve and Bayes, Which can be described as:

- **Naïve**: It assumes that the features are independent of occurrence of other while predicting the target feature which is Naive as often is not possible all the time which might lead to incorrect results. If we are predicting is a fruit, then color, shape, taste would be required to identify it. If red, round and sweet is our data point then it will be identified as Apple.
- **Bayes**: It is called Bayes because it depends on the principle of Bayes Theorem which we will look in next section.

Apart from considering the independence of every feature, Naive Bayes also assumes that they contribute equally. This is an important point to remember.

> *Bayes' Theorem finds the probability of an event occurring given the probability of another event that has already occurred. Bayes' theorem is stated mathematically as the following equation:*


Probability of occurrence of event X, given event Y has already occurred
Probability of occurrence of event Y
Probability of occurrence of event Y, given event X has already occurred
P(Y|X) = P(X|Y) * P(Y) / P(X)
Probability of occurrence of event X

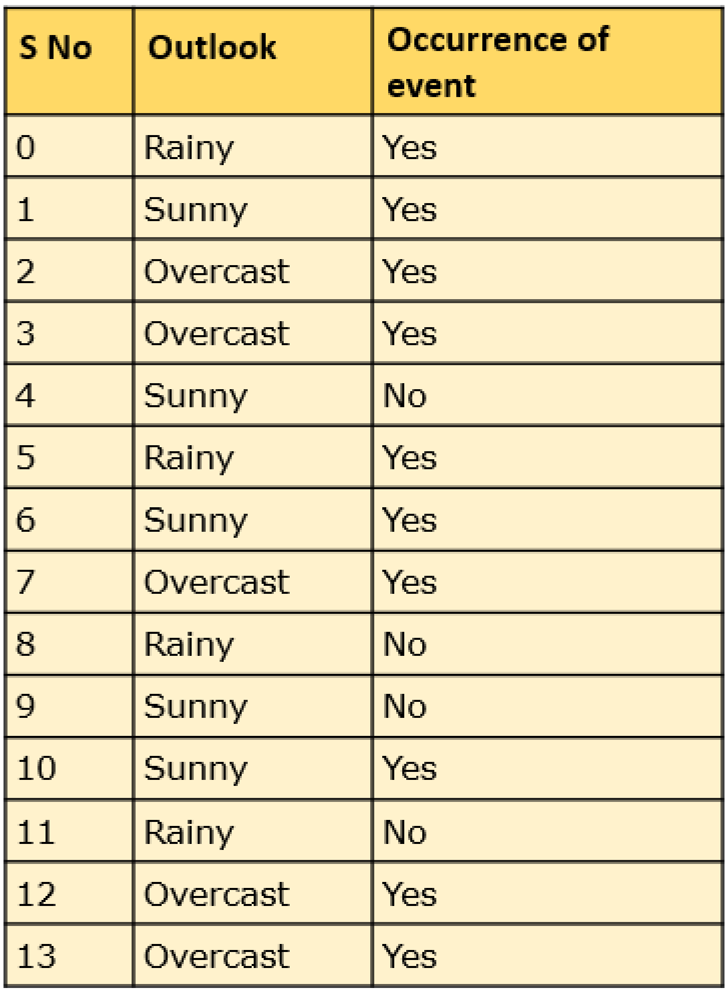
Bayes Theorem

- Basically, we are trying to find probability of event Y, given the event X is true.
- P(X|Y) is called **Likelihood** which is is the probability of *predictor* given *class*.
- Event X is also termed as **evidence** so P(X) is the probability of evidence. P(Y) is the **priori** of Y (the prior probability, i.e. Probability of event before evidence is seen). The evidence is an attribute value of an unknown instance(here, it is event X).
- P(Y|X) is **posteriori probability** of X, i.e. probability of event after evidence is seen.

We have to find the posterior probability or P(Y|X). For multiple values of Y, we will need to calculate this expression for each of them. Given a new instance Xnew, we need to calculate the probability that Y will take on any given value, given the observed attribute values of Xnew and given the distributions P(Y) and P(X|Y) estimated from the training data.

To predict the class of the target variable, based on the different values of P(Y|X), the algorithm takes the maximum of these probability values. This is also called **maximizing a posteriori(MAP)**.

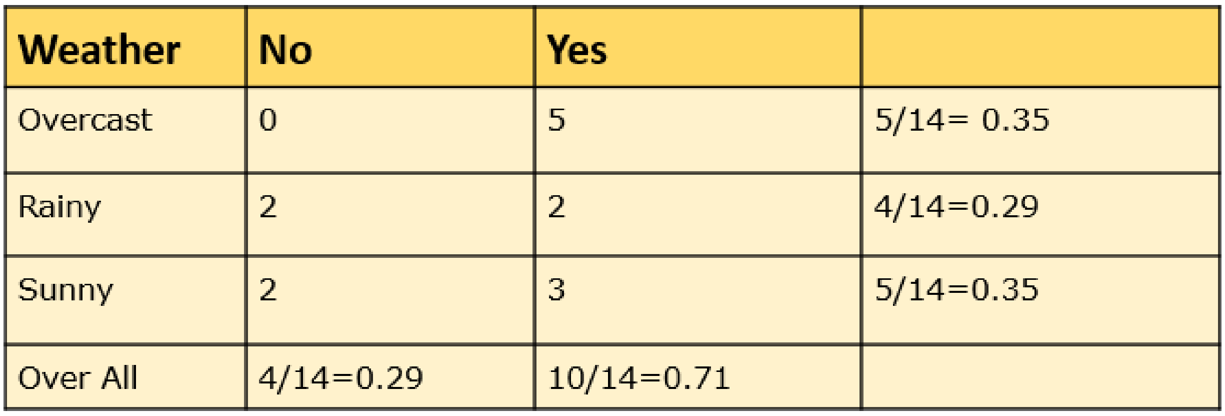
Lets look at an example to understand this better:



| S No | Outlook | Occurrence of event |
|---|---|---|
| 0 | Rainy | Yes |
| 1 | Sunny | Yes |
| 2 | Overcast | Yes |
| 3 | Overcast | Yes |
| 4 | Sunny | No |
| 5 | Rainy | Yes |
| 6 | Sunny | Yes |
| 7 | Overcast | Yes |
| 8 | Rainy | No |
| 9 | Sunny | No |
| 10 | Sunny | Yes |
| 11 | Rainy | No |
| 12 | Overcast | Yes |
| 13 | Overcast | Yes |

Example with data which gives condition of whether and if the player played or not

If we derive the frequency of occurrence of each event of this data and calculate the likelihoods, it looks like



| Weather | No | Yes | |
|---|---|---|---|
| Overcast | 0 | 5 | 5/14= 0.35 |
| Rainy | 2 | 2 | 4/14=0.29 |
| Sunny | 2 | 3 | 5/14=0.35 |
| Over All | 4/14=0.29 | 10/14=0.71 | |

Likelihood table

**Applying Bayes' theorem:**

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

P(Sunny|Yes)= 3/10= 0.3, P(Sunny) = 0.35, P(Yes)=0.71

=> P(Yes|Sunny) = 0.3*0.71/0.35= **0.60**

**P(No|Sunny)= P(Sunny|No)*P(No)/P(Sunny)**

P(Sunny|NO)= 2/4=0.5 , P(No)= 0.29, P(Sunny)= 0.35

=> P(No|Sunny)= 0.5*0.29/0.35 = **0.41**

So as we can see from the above calculation that
**P(Yes|Sunny)>P(No|Sunny) Hence on a Sunny day, Player can play the game.**

As we have only one feature, we have one likelihood table, as the features increase, the same concept can be applied and result can be inferred from final calculations.

X = x1, x2, x3, …., xn. Here, x1, x2,…, xn stand for the features. Now, Bayes equation will be changed to

**P(y | x1, …, xn) = [P(x1 | y) P(x2 | y) … P(xn | y) P(y)]/[P(x1) P (x2) … P(xn)].**

## Feature Distribution

As seen above, we need to estimate the distribution of the response variable from training set or assume uniform distribution. Similarly, ***to estimate the parameters for a feature's distribution, one must assume a distribution or generate nonparametric models for the features from the training set***. Such assumptions are **known as event models.** The variations in these assumptions generates different algorithms for different purposes. **For continuous distributions, the Gaussian naive Bayes is the algorithm of choice. For discrete features, multinomial and Bernoulli distributions as popular**.

- **Gaussian**: The likelihood of the features is assumed to be Gaussian. The Gaussian model assumes that all features fall in normal distribution. This means if predictors take continuous values instead of discrete, then the model assumes that these values are sampled from the Gaussian distribution.

$$P(x_i \mid y) = \frac{1}{\sqrt{2\pi\sigma_y^2}} \exp\left(-\frac{(x_i - \mu_y)^2}{2\sigma_y^2}\right)$$

The parameters σy and μy are estimated using maximum likelihood

- **Multinomial**: The Multinomial Naïve Bayes classifier is used when the data is multinomial distributed. It is primarily used for document classification problems. The distribution is parametrized by vectors θy=(θy1,…,θyn) for each class y, where n is the number of features (in text classification, the size of the vocabulary) and θyi is the probability P(xi|y) of feature i appearing in a sample belonging to class y. The parameters θy is estimated by a smoothed version of maximum likelihood, i.e. relative frequency counting:

$$\hat{\theta}_{yi} = \frac{N_{yi} + \alpha}{N_y + \alpha n}$$

source sklearn

where Nyi=∑x∈Txi is the number of times feature i appears in a sample of class y in the training set T, and Ny=∑i=1nNyi is the total count of all features for class y.

The smoothing priors α≥0 accounts for features not present in the learning samples and prevents zero probabilities in further computations. Setting α=1 is called Laplace smoothing, while α<1 is called Lidstone smoothing.

- **Bernoulli**: The Bernoulli classifier works similar to the Multinomial classifier, but the predictor variables are the independent Booleans variables. Such as if a particular word is present or not in a document. This model is also famous for document classification tasks. The rule in Bernoulis theorem would be

$$P(x_i \mid y) = P(i \mid y)x_i + (1 - P(i \mid y))(1 - x_i)$$

Along with these, there are few other types of functions available for Naive Bayes. We can pick which type of classifier from sklearn.

## Advantages and Disadvantages of Naive Bayes

### Advantages

- As mentioned this algorithm works fast
- **Naive Bayes** is suitable for solving multi-class prediction problems.
- If its assumption of the independence of features holds true, it can perform better than other models and requires much less training data.
- **Naive Bayes** is better suited for categorical input variables than numerical variables. Because of this reason it is used in text classification tasks.

### Disadvantages

- Naive Bayes assumes that all predictors (or features) are independent. In real world data it is not observed very frequently.
- 'zero-frequency problem': If test data set has a categorical variable whose category wasn't available in the training dataset, then it assigns *zero-probability*

Naive Bayes Naive Bayes Classifier Naive Bayes From Scratch Classification