The Naïve Bayes Algorithm

Naïve Bayes is a classification algorithm based on **Bayes' theorem** that *naïvely* assumes independence between features and assigns the same weight (degree of importance) to all features. This means that the algorithm assumes that no single feature correlates to or affects another. For example, although weight and height are somehow correlated when predicting a person's age, the algorithm assumes that each feature is independent. Additionally, the algorithm considers all features equally important. For instance, even though an education degree may influence the earnings of a person to a greater degree than the number of children the person has, the algorithm still considers both features equally important.

Note

Bayes' theorem is a mathematical formula that calculates conditional probabilities. To learn more about this theorem, visit the following URL: *https://plato.stanford.edu/entries/bayes-theorem/*.

Although real-life datasets contain features that are not equally important, nor independent, this algorithm is popular among scientists as it performs

surprisingly well on large datasets. Also, due to the simplistic approach of the algorithm, it runs quickly, thus allowing it to be applied to problems that require predictions in real-time. Moreover, it is frequently used for text classification as it commonly outperforms more complex algorithms.

How Does the Naïve Bayes Algorithm Work?

The algorithm converts the input data into a summary of occurrences of each class label against each feature, which is then used to calculate the likelihood of one event (a class label), given a combination of features. Finally, this likelihood is normalized against the likelihood of the other class labels. The result is the probability of an instance belonging to each class label. The sum of the probabilities must be one, and the class label with a higher probability is the one that the algorithm chooses as the prediction.

Let's take, for example, the data presented in the following tables:

A		
Weather	**Temperature**	**Outcome**
Sunny	Hot	Yes
Sunny	Cool	Yes
Rainy	Cold	No
Sunny	Hot	No
Mild	Cool	Yes
Mild	Cool	Yes
Sunny	Hot	Yes
Rainy	Cool	No
Rainy	Cold	Yes
Sunny	Hot	Yes

B		
Weather	**Temperature**	**Outcome**
Sunny	4	1
Rainy	1	2
Mild	2	0

Temprature	**Yes**	**No**
Hot	3	1
Cool	3	1
Cold	1	1

Overall	**Yes**	**No**
	7	3

Figure 2: Table A - Input data and Table B - Occurrence count

Table A represents the data that is fed to the algorithm to build the model. Table B refer to the occurrence count that the algorithm uses implicitly to calculate the probabilities.

To calculate the likelihood of an event occurring when given a set of features, the algorithm multiplies the probability of the event occurring, given each individual feature, by the probability of the occurrence of the event, independent of the rest of the features, as follows:

```
Likelihood [A1|E] = P[A1|E1] * P[A1|E2] * … * P[A1|En] * P[A1]
```

Here, A_1 refers to an event (one of the class labels) and E represents the set of features, where E_1 is the first feature and E_n is the last feature in the dataset.

Note

The multiplication of these probabilities can only be made by assuming independence between features.

The preceding equation is calculated for all possible outcomes (all class labels), and then the normalized probability of each outcome is calculated as follows:

$$P[A_1|E] = \frac{likelihood[A_1|E]}{likelihood[A_1|E] + likelihood[A_2|E] + \cdots + likelihood[A_n|E]}$$

Figure 3: Formula to calculate normalized probability

For the example in *Figure 2*, given a new instance with weather equal to *sunny* and temperature equal to *cool*, the calculation of probabilities is as follows:

$$Likelihood[yes|sunny, cool] = \frac{4}{7} * \frac{3}{7} * \frac{7}{10} = 0.17$$

$$Likelihood[no|sunny, cool] = \frac{1}{3} * \frac{1}{3} * \frac{3}{10} = 0.03$$

$$P[yes|sunny, cool] = \frac{0.17}{0.17 + 0.03} = 0.85 \approx 85\%$$

$$P[no|sunny, cool] = \frac{0.03}{0.17 + 0.03} = 0.15 \approx 15\%$$

Figure 4: Calculation of the likelihood and probabilities for the example dataset

By looking at the preceding equations, it is possible to conclude that the prediction should be *yes*.

It is important to mention that for continuous features, the summary of occurrences is done by creating ranges. For instance, for a feature of price, the algorithm may count the number of instances with prices below 100K, as well as the instances with prices above 100K.

Moreover, the algorithm may encounter some issues if one value of a feature is never associated with one of the outcomes. This is an issue mainly because the probability of the outcome given that feature will be zero, which influences the entire calculation. In the preceding example, for predicting the outcome of an instance with weather equal to *mild* and temperature equal to *cool*, the probability of *no*, given the set of features will be equal to zero, considering that the probability of *no*, given *mild* weather, computes to zero, since there are no occurrences of *mild* weather when the outcome is *no*.

To avoid this, the **Laplace estimator** technique should be used. Here, the fractions representing the probability of the occurrence of an event given a feature, $P[A|E_1]$, are modified by adding 1 to the numerator while also adding the number of possible values of that feature to the denominator.

For this example, to perform a prediction for a new instance with weather equal to *mild* and temperature equal to *cool* using the Laplace estimator, this would be done as follows:

$$Likelihood[yes|mild,cool] = \frac{3}{10} * \frac{4}{10} * \frac{7}{10} = 0.084$$

$$Likelihood[no|mild,cool] = \frac{1}{6} * \frac{2}{6} * \frac{3}{10} = 0.016$$

$$P[yes|mild,cool] = \frac{0.084}{0.084 + 0.016} = 0.84 \approx 84\%$$

$$P[no|mild,cool] = \frac{0.016}{0.084 + 0.016} = 0.16 \approx 16\%$$

Figure 5: Calculation of the likelihood and probability using the Laplace estimator for the example dataset

Here, the fraction that calculates the occurrences of *yes*, given *mild* weather, goes from 2/7 to 3/10, as a result of the addition of 1 to the numerator and 3 (for *sunny*, *mild*, and *rainy*) to the denominator. The same goes for the other fractions that calculate the probability of the event, given a feature. Note that the fraction that calculates the probability of the event occurring independently of any feature is left unaltered.

Nevertheless, as you have learned so far, the scikit-learn library allows you to train models and then

use them for predictions, without needing to hardcode the math.