05

Classification: Naïve Bayes

Naïve Bayes

Naïve Bayes is the simplest algorithm that you can apply to your data. As the name suggests, here this algorithm makes an assumption as all the variables in the dataset are “Naïve” i.e not correlated to each other.

Naive Bayes is a probabilistic machine learning algorithm based on the Bayes Theorem, used in a wide variety of classification tasks.

Bayes Theorem

Named after Thomas Bayes from the 1700s. The Naive Bayes classifier works on the principle of conditional probability, as given by the Bayes theorem.

$$P(A|B) = P(A) \times \frac{P(B|A)}{P(B)}$$

posterior: $P(A|B)$ — prior: $P(A)$ — likelihood: $P(B|A)$ — marginal: $P(B)$

- The Bayes theorem gives us the conditional probability of event A, given that event B has occurred.
- Posterior probability (updated probability after the evidence is considered)
- Prior probability (the probability before the evidence is considered)
- Likelihood (probability of the evidence, given the belief is true)
- Marginal probability (probability of the evidence, under any circumstance)

Bayes' Theorem Explained with 4 number-table

Imagine 100 people at a party, and you tally how many wear pink or not, and if a man or not, and get these numbers:

	Pink	notPink
Man	5	35
notMan	20	40

	Pink	notPink	
Man	5	35	40
notMan	20	40	60
	25	75	100

- the probability of being a man is P(Man) = 40/100 = 0.4
- the probability of wearing pink is P(Pink) = 25/100 = 0.25
- the probability that a man wears pink is P(Pink|Man) = 5/40 = 0.125
- the probability that a person wearing pink is a man P(Man|Pink) = ...

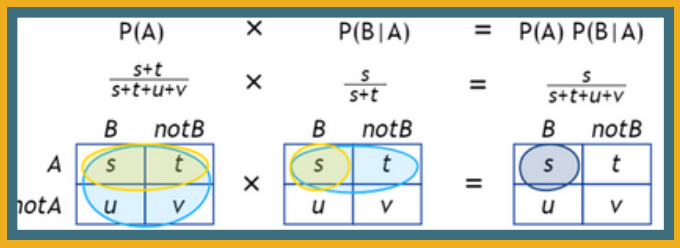
$$P(Man|Pink) = \frac{P(Man)P(Pink|Man)}{P(Pink)} = \frac{0.4 \times 0.125}{0.25} = 0.2$$ Directly from table → $\frac{5}{25} = 0.2$

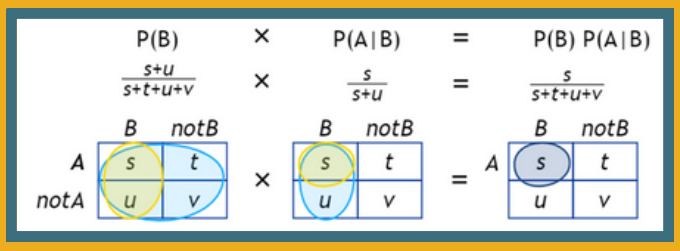
Generalization of 4-number table

P(B)P(A|B)= P(A)P(B|A)

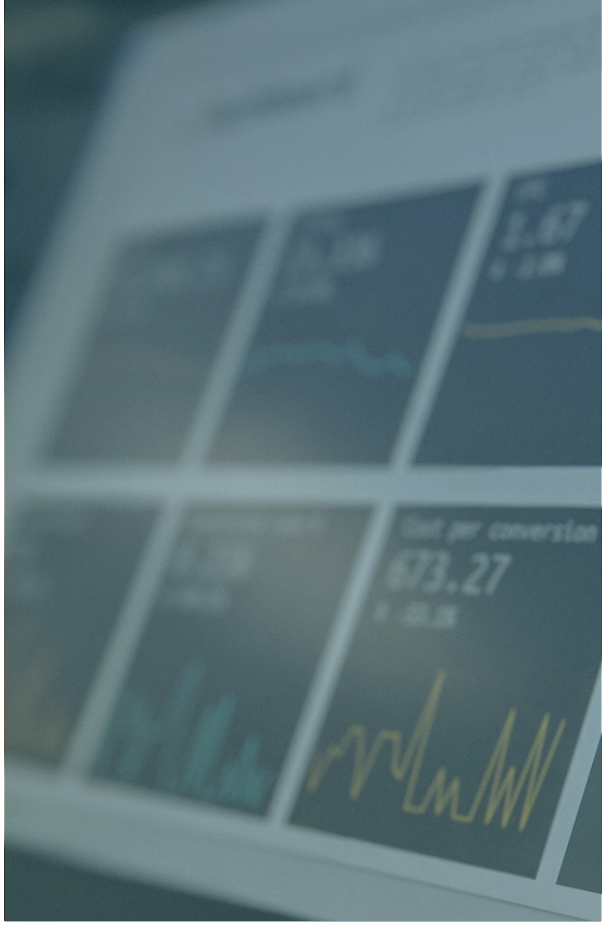
If there is no P(B), find it using

P(B)=P(A) P(B|A)+P(not A) P(B|not A)





Naive Bayes Classifier



Naive Bayes does seem to be a simple yet powerful algorithm. But why is it so popular? Since it is a probabilistic model, the algorithm can be coded up easily and the predictions made real quick. Real-time quick. Because of this, it is easily scalable and is traditionally the algorithm of choice for real-world applications (apps) that are required to respond to user's requests instantaneously.

The fundamental Naive Bayes assumption is that each feature makes an independent and equal contribution to the outcome. With relation to our dataset, this concept can be understood as:

- We assume that no pair of features are dependent. For example, the temperature being 'Hot' has nothing to do with the humidity or the outlook being 'Rainy' has no effect on the winds. Hence, the features are assumed to be independent.
- Secondly, each feature is given the same weight(or importance). For example, knowing only temperature and humidity alone can't predict the outcome accurately. None of the attributes is irrelevant and assumed to be contributing equally to the outcome.

Naïve Bayes Algorithm

Day	Outlook	Temp.	Humidity	Windy	Play Golf
D1	Rainy	Hot	High	FALSE	No
D2	Rainy	Hot	High	TRUE	No
D3	Overcast	Hot	High	FALSE	Yes
D4	Sunny	Mild	High	FALSE	Yes
D5	Sunny	Cool	Normal	FALSE	Yes
D6	Sunny	Cool	Normal	TRUE	No
D7	Overcast	Cool	Normal	TRUE	Yes
D8	Rainy	Mild	High	FALSE	No
D9	Rainy	Cool	Normal	FALSE	Yes
D10	Sunny	Mild	Normal	FALSE	Yes
D11	Rainy	Mild	Normal	TRUE	Yes
D12	Overcast	Mild	High	TRUE	Yes
D13	Overcast	Hot	Normal	FALSE	Yes
D14	Sunny	Mild	High	TRUE	No

Now, with regards to our dataset, we can apply Bayes' theorem in following way:

$$P(y|X) = \frac{P(X|y)P(y)}{P(X)}$$

Where, y is class variable and X is a dependent feature vector (of size n) where:

$$X = (x_1, x_2, x_3, \ldots, x_n)$$

Just to be clear, an example of a feature vector and corresponding class variable can be: (refer 1st row of dataset)

x = (Rainy, Hot, High, False)
y = No

So basically, P(y|X) here means, the probability of "Not playing golf" given that the weather conditions are "Rainy outlook", "Temperature is hot", "high humidity" and "no wind".

Outlook

	Yes	No	P(yes)	P(no)
Sunny	2	3	2/9	3/5
Overcast	4	0	4/9	0/5
Rainy	3	2	3/9	2/5
Total	9	5	100%	100%

Temperature

	Yes	No	P(yes)	P(no)
Hot	2	2	2/9	2/5
Mild	4	2	4/9	2/5
Cool	3	1	3/9	1/5
Total	9	5	100%	100%

Humidity

	Yes	No	P(yes)	P(no)
High	3	4	3/9	4/5
Normal	6	1	6/9	1/5
Total	9	5	100%	100%

Wind

	Yes	No	P(yes)	P(no)
False	6	2	6/9	2/5
True	3	3	3/9	3/5
Total	9	5	100%	100%

Play		P(Yes)/P(No)
Yes	9	9/14
No	5	5/14
Total	14	100%

So, in the frequency tables, we have calculated P(xi | yj) for each xi in X and yj in y manually. For example, probability of temperature is cool given that playing golf is true: P(CoolTempl Yes) = 3/9.

Naïve Bayes Algorithm

Let us test a new set of features (calledtoday): today = (Sunny, Hot, Normal, False)

$$P(Yes|today) = \frac{P(Yes)\,P(SunnyOutlook|Yes)P(HotTemp|Yes)P(NormalHumidity|Yes)P(NoWind|Yes)}{P(today)}$$

$$P(No|today) = \frac{P(No)\,P(SunnyOutlook|No)P(HotTemp|No)P(NormalHumidity|No)P(NoWind|No)}{P(today)}$$

Since, P(today) is common in both probabilities, we can ignore P(today) and find proportional probabilities

$$P(Yes|today) = \frac{9}{14}\frac{2}{9}\frac{2}{9}\frac{6}{9}\frac{6}{9} \approx 0.0141$$

$$P(No|today) = \frac{5}{14}\frac{3}{5}\frac{2}{5}\frac{1}{5}\frac{2}{5} \approx 0.0068$$

P (Yes|today) > P (No|today)
So, prediction that golf would be played is 'Yes'

Laplace Correction

If you look at the frequency table, you'll find that P(OvercastOutlook|No) equals to 0. The implication is that when we want to predict where the Outlook value is overcast, the whole calculation is nullified.

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'.

Gaussian Naïve Bayes

We've seen the computations when the X's are categorical. But how to compute the probabilities when X is a continuous variable?
If we assume that the X follows a particular distribution, then you can plug in the probability density function of that distribution to compute the probability of likelihoods.
If the likelihood of the features is assumed to be Gaussian, hence, conditional probability is given by:

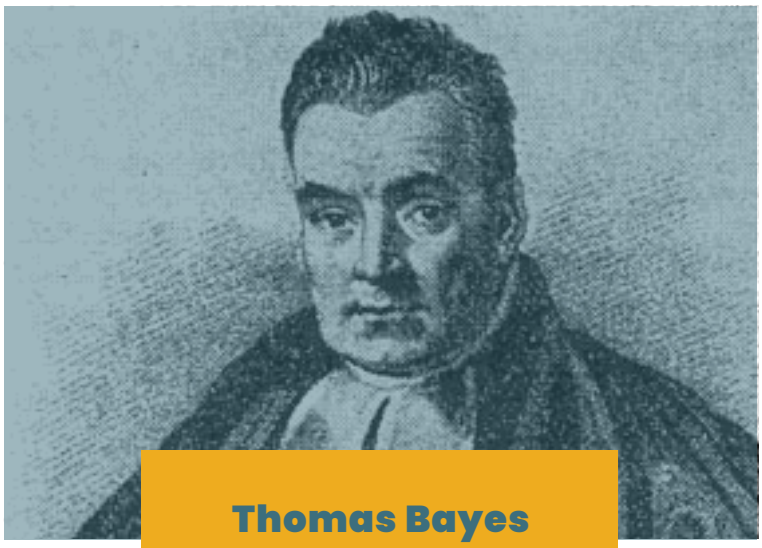
$$P(X|Y = c) = \frac{1}{\sqrt{2\pi\sigma_c^2}} e^{\frac{-(x-\mu_c)^2}{2\sigma_c^2}}$$

Other Popular Naïve Bayes Classifiers

- Multinomial Naive Bayes: Feature vectors represent the frequencies with which certain events have been generated by a multinomial distribution. This is the event model typically used for document classification.
- Bernoulli Naive Bayes: In the multivariate Bernoulli event model, features are independent booleans (binary variables) describing inputs. Like the multinomial model, this model is popular for document classification tasks, where binary term occurrence (i.e., a word occurs in a document or not) features are used rather than term frequencies (i.e., frequency of a word in the document).

Advantages of Naïve Bayes Classifier

- It is simple and easy to implement
- It doesn't require as much training data
- It handles both continuous and discrete data
- It is highly scalable with the number of predictors and data points
- It is fast and can be used to make real-time predictions
- It is not sensitive to irrelevant features



Thomas Bayes

Disadvantages of Naïve Bayes Classifier

- If your test data set has a categorical variable of a category that wasn't present in the training data set, the Naive Bayes model will assign it zero probability and won't be able to make any predictions in this regard. This phenomenon is called 'Zero Frequency,' and you'll have to use a smoothing technique to solve this problem.
- 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.
- It assumes that all the features are independent. While it might sound great in theory, in real life, you'll hardly find a set of independent features.

Tips to Improve Power of Naive Bayes Model

- If continuous features do not have normal distribution, we should use transformation or different methods to convert it into normal distribution.
- If the test data set has zero frequency issue, apply smoothing techniques "Laplace Correction" to predict the class of the test data set.
- Remove correlated features, as the highly correlated features are voted twice in the model and it can lead to over inflating importance.
- Naive Bayes classifiers have limited options for parameter tuning. I would recommend to focus on your pre-processing of data and the feature selection.
- You might think to apply some classifier combination techniques like ensembling, bagging and boosting but these methods would not help. Actually, "ensembling, boosting, bagging" won't help since their purpose is to reduce variance. Naive Bayes has no variance to minimize.

Picture Source

- pexels.com
- https://upload.wikimedia.org/wikipedia/commons/d/d4/Thomas_Bayes.gif