Naive Bayes Classifier: Learning Naive Bayes with Python

What is Naive Bayes?

Naive Bayes is among one of the most simple and powerful algorithms for **classification** based on Bayes' Theorem with an assumption of independence among predictors. Naive Bayes model is easy to build and particularly useful for very large data sets. There are two parts to this algorithm:

- **Naive**
- **Bayes**

The Naive Bayes classifier assumes that the presence of a feature in a class is unrelated to any other feature. Even if these features depend on each other or upon the existence of the other features, all of these properties independently contribute to the probability that a particular fruit is an apple or an orange or a banana and that is why it is known as **"Naive".**

What is Bayes Theorem?

In Statistics and probability theory, Bayes' theorem describes the probability of an event, based on prior knowledge of conditions that might be related to the event. It serves as a way to figure out conditional probability.

Given a Hypothesis **H** and evidence **E,** Bayes' Theorem states that the relationship between the probability of Hypothesis before getting the evidence **P(H)** and the probability of the hypothesis after getting the evidence **P(H|E)** is :

$$P(H|E) = \frac{P(E|H).P(H)}{P(E)}$$

This relates the probability of the hypothesis before getting the evidence **P(H)**, to the probability of the hypothesis after getting the evidence, **P(H|E)**. For this reason, is called the **prior probability**, while **P(H|E)** is called the **posterior probability**. The factor that relates the two, **P(H|E) / P(E)**, is called the **likelihood ratio**. Using these terms, Bayes' theorem can be rephrased as:

“The posterior probability equals the prior probability times the likelihood ratio.”

Bayes’ Theorem Example

Let’s suppose we have a Deck of Cards, we wish to find out the “**Probability of the Card we picked at random to be a King given that it is a Face Card**“. So, according to Bayes Theorem, we can solve this problem. First, we need to find out the probability

- **P(King)** which is **4/52** as there are 4 Kings in a Deck of Cards.
- **P(Face|King)** is equal to **1** as all the Kings are face Cards.
- **P(Face)** is equal to **12/52** as there are 3 Face Cards in a Suit of 13 cards and there are 4 Suits in total.

$$P(King|Face) = \frac{P(Face|King).P(King)}{P(Face)}$$

$$= \frac{1.(1/13)}{3/13} = 1/3$$

P(King) = 4/52 =1/13

P(Face|King) = 1

P(Face) =12/52 = 3/13

Now, putting all the values in the Bayes’ Equation we get the result as **1/3**

Game Prediction using Bayes’ Theorem

Let’s continue our Naive Bayes Tutorial blog and Predict the Future of Playing with the weather data we have.

So here we have our Data, which comprises of the Day, Outlook, Humidity, Wind Conditions and the final column being Play, which we have to predict.

Day	Outlook	Humidity	Wind	Play
D1	Sunny	High	Weak	No
D2	Sunny	High	Strong	No
D3	Overcast	High	Weak	Yes
D4	Rain	High	Weak	Yes
D5	Rain	Normal	Weak	Yes
D6	Rain	Normal	Strong	No
D7	Overcast	Normal	Strong	Yes
D8	Sunny	High	Weak	No
D9	Sunny	Normal	Weak	Yes
D10	Rain	Normal	Weak	Yes
D11	Sunny	Normal	Strong	Yes
D12	Overcast	High	Strong	Yes
D13	Overcast	Normal	Weak	Yes
D14	Rain	High	Strong	No

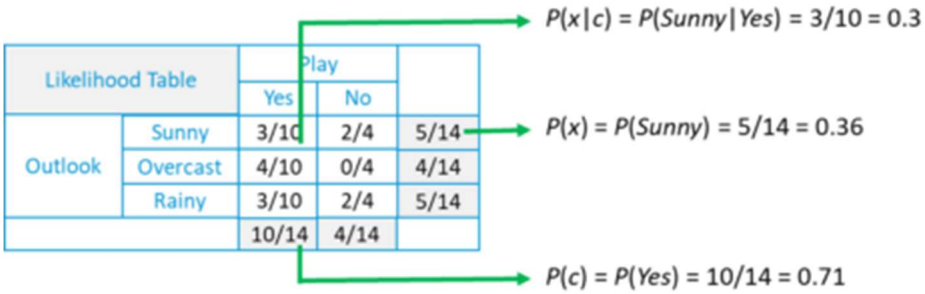
- First, we will create a **frequency** table using each attribute of the dataset.

Frequency Table		Play	
		Yes	No
Outlook	Sunny	3	2
	Overcast	4	0
	Rainy	3	2

Frequency Table		Play	
		Yes	No
Humidity	High	3	4
	Normal	6	1

Frequency Table		Play	
		Yes	No
Wind	Strong	6	2
	Weak	3	3

- For each frequency table, we will generate a **likelihood** table.



Likelihood Table		Play		
		Yes	No	
Outlook	Sunny	3/10	2/4	5/14
	Overcast	4/10	0/4	4/14
	Rainy	3/10	2/4	5/14
		10/14	4/14	

- Likelihood of '**Yes**' given '**Sunny**' is:

P(c|x) = P(Yes|Sunny) = P(Sunny|Yes)* P(Yes) / P(Sunny) = (0.3 x 0.71) /0.36 = 0.591

- Similarly Likelihood of '**No**' given '**Sunny**' is:

P(c|x) = P(No|Sunny) = P(Sunny|No)* P(No) / P(Sunny) = (0.4 x 0.36) /0.36 = 0.40

- Now, in the same way, we need to create the Likelihood Table for other attributes as well.

Likelihood table for Humidity

Likelihood Table		Play		
		Yes	No	
Humidity	High	3/9	4/5	7/14
	Normal	6/9	1/5	7/14
		9/14	5/14	

$P(Yes|High)$ = 0.33 x 0.6 / 0.5 = 0.42

$P(No|High)$ = 0.8 x 0.36 / 0.5 = 0.58

Likelihood table for Wind

Likelihood Table		Play		
		Yes	No	
Wind	Weak	6/9	2/5	8/14
	Strong	3/9	3/5	6/14
		9/14	5/14	

$P(Yes|Weak)$ = 0.67 x 0.64 / 0.57 = 0.75

$P(No|Weak)$ = 0.4 x 0.36 / 0.57 = 0.25

Suppose we have a **Day** with the following values :

- **Outlook = Rain**
- **Humidity = High**
- **Wind = Weak**

- **Play =?**

- So, with the data, we have to predict whether “we can play on that day or not”.

Likelihood of ‘Yes’ on that Day = P(Outlook = Rain|Yes)*P(Humidity= High|Yes)* P(Wind= Weak|Yes)*P(Yes)

= 2/9 * 3/9 * 6/9 * 9/14 = 0.0199

Likelihood of ‘No’ on that Day = P(Outlook = Rain|No)*P(Humidity= High|No)* P(Wind= Weak|No)*P(No)

= 2/5 * 4/5 * 2/5 * 5/14 = 0.0166

- Now we normalize the values, then

P(Yes) = 0.0199 / (0.0199+ 0.0166) = 0.55

P(No) = 0.0166 / (0.0199+ 0.0166) = 0.45

- Our model predicts that there is a **55%** chance there will be a Game tomorrow.

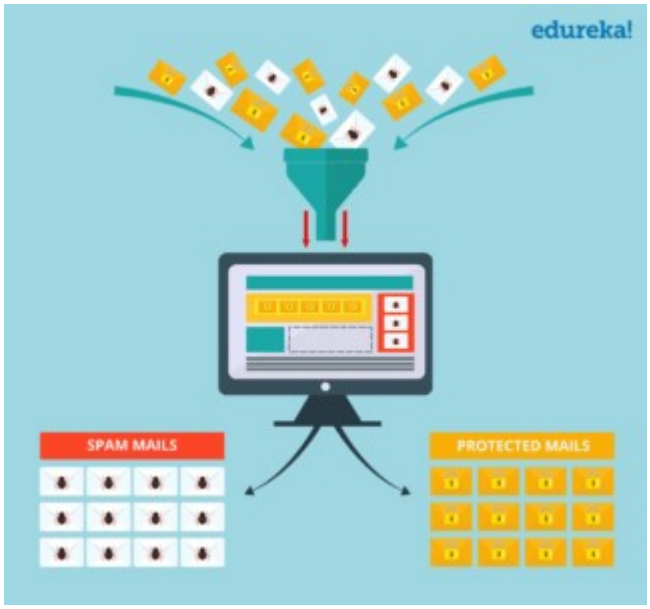
Naive Bayes in the Industry

Now that you have an idea of What exactly is Naïve Bayes, how it works, let’s see where is it used in the Industry?

News Categorization:

Starting with our first industrial use, it is **News Categorization,** or we can use the term text classification to broaden the spectrum of this algorithm. News on the web is rapidly growing where each news site has its own different layout and categorization for grouping news. Companies use a web crawler to extract useful text from HTML pages of news article contents to construct a Full-Text-RSS. Each news article contents is **tokenized**(categorized). In order to achieve better classification result, we remove the less significant words i.e. stop – word from the document. We apply the naive Bayes classifier for classification of news contents based on news code.

Spam Filtering:



Naive Bayes classifiers are a popular statistical technique of e-mail filtering. They typically use a bag of words features to identify spam e-mail, an approach commonly used in text classification. Naive Bayes classifiers work by correlating the use of tokens (typically words, or sometimes other things), with a spam and non-spam e-mails and then using Bayes' theorem to calculate a probability that an email is or is not spam.

Particular words have particular probabilities of occurring in spam email and in legitimate email. For instance, most email users will frequently encounter the word "Lottery" and "Luck Draw" in spam email, but will seldom see it in other emails. Each word in the email contributes to the email's spam probability or only the most interesting words. This contribution is called the **posterior probability** and is computed using **Bayes' theorem.** Then, the email's spam probability is computed over all words in the email, and if the total exceeds a certain threshold (say 95%), the filter will mark the email as a spam.

Medical Diagnosis:

Nowadays modern hospitals are well equipped with monitoring and other data collection devices resulting in enormous data which are collected continuously through health examination and medical treatment. One of the main advantages of the Naive Bayes approach which is appealing to physicians is that **"all the available information is used to explain the decision"**. This explanation seems to be "natural" for medical diagnosis and prognosis i.e. is close to the way how physicians diagnose patients.

When dealing with medical data, Naïve Bayes classifier takes into account evidence from many attributes to make the final prediction and provides transparent explanations of its decisions and therefore it is considered as one of the most useful classifiers to support physicians' decisions.

Weather Prediction:

Weather is one of the most influential factors in our daily life, to an extent that it may affect the economy of a country that depends on occupation like agriculture. Weather prediction has been a challenging problem in the meteorological department for years. Even after the technological and scientific advancement, the accuracy in prediction of weather has never been sufficient.

A Bayesian approach based model for weather prediction is used, where **posterior probabilities** are used to calculate the **likelihood** of each class label for input data instance and the one with **maximum likelihood** is considered resulting output.\

Here we have a dataset comprising of 768 Observations of women aged 21 and older. The dataset describes instantaneous measurement taken from patients, like age, blood workup, the number of times pregnant. Each record has a class value that indicates whether the patient suffered an onset of diabetes within 5 years. The values are **1** for Diabetic and **0** for Non-Diabetic.