Naive Bayes Algorithm

- Naive Bayes is a probabilistic machine learning algorithm that can be used in a wide variety of classification tasks.
- Typical applications include filtering spam, classifying documents, sentiment prediction etc.
- It is based on the works of Rev. Thomas Bayes (1702–61) and hence the name.

But why is it called ‘Naive’?

- The name naive is used because it assumes the features that go into the model is independent of each other. That is changing the value of one feature, does not directly influence or change the value of any of the other features used in the algorithm.

Naive Bayes does seem to be a simple yet powerful algorithm. But why is it so popular?

- That's because there is a significant advantage with NB.
- 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.

But before you go into Naive Bayes, you need to understand what ‘Conditional Probability’ is and what is the ‘Bayes Rule’.

And by the end of this SESSION, you will know:

- How exactly Naive Bayes Classifier works step-by-step
- What is Gaussian Naive Bayes, when is it used and how it works?
- How to code it up in Python
- How to improve your Naive Bayes models?

What is Conditional Probability?

- Mathematically, Conditional probability of A given B can be computed as: P(A|B) = P(A AND B) / P(B)

School Example

- Consider a school with a total population of 100 persons. These 100 persons can be seen either as ‘Students’ and ‘Teachers’ or as a population of ‘Males’ and ‘Females’.
- With below tabulation of the 100 people, what is the conditional probability that a certain member of the school is a ‘Teacher’ given that he is a ‘Man’?

	Female	Male	Total
Teacher	8	12	20
Student	32	48	80
Total	40	60	100

To calculate this, you may intuitively filter the sub-population of 60 males and focus on the 12 (male) teachers.
So the required conditional probability P(Teacher | Male) = 12 / 60 = 0.2.

$$P(Teacher \mid Male) = \frac{P(Teacher \cap Male)}{P(Male)} = 12/60 = 0.2$$

- This can be represented as the intersection of Teacher (A) and Male (B) divided by Male (B). Likewise, the conditional probability of B given A can be computed. The Bayes Rule that we use for Naive Bayes, can be derived from these two notations.

$$P(A \mid B) = \frac{P(A \cap B)}{P(B)} \qquad (1)$$

$$P(B \mid A) = \frac{P(A \cap B)}{P(A)} \qquad (2)$$

The Bayes Rule

- The Bayes Rule is a way of going from P(X|Y), known from the training dataset, to find P(Y|X).

For observations in test or scoring data, the X would be known while Y is unknown. And for each row of the test dataset, you want to compute the probability of Y given the X has already happened.

What happens if Y has more than 2 categories? we compute the probability of each class of Y and let the highest win.

Bayes Rule is a way to go from P (X | Y) to find P (Y | X)

$$P(X \mid Y) = \frac{P(X \cap Y)}{P(Y)} \quad (1)$$

Known

$$P(Y \mid X) = \frac{P(X \cap Y)}{P(X)} \quad (2)$$

UnKnown

P (Evidence | Outcome)
(Known from training data)

P (Outcome | Evidence)
(To be predicted for test data)

Bayes Rule

$$P(Y \mid X) = \frac{P(X \mid Y) * P(Y)}{P(X)}$$

The Naive Bayes

- The Bayes Rule provides the formula for the probability of Y given X. But, in real-world problems, you typically have multiple X variables.
- When the features are independent, we can extend the Bayes Rule to what is called Naive Bayes.
- It is called ‘Naive’ because of the naive assumption that the X’s are independent of each other. Regardless of its name, it’s a powerful formula.

When there are multiple X variables, we simplify it by *assuming the X's are independent*, so the **Bayes** rule

$$P(Y=k \mid X) = \frac{P(X \mid Y=k) * P(Y=k)}{P(X)}$$

where, k is a class of Y

becomes, Naive **Bayes**

$$P(Y=k \mid X1..Xn) = \frac{P(X1 \mid Y=k) * P(X2 \mid Y=k) \ldots * P(Xn \mid Y=k) * P(Y=k)}{P(X1) * P(X2) \ldots * P(Xn)}$$

$$P(Y=k \mid X1..Xn) = \frac{P(X1 \mid Y=k) * P(X2 \mid Y=k) \ldots * P(Xn \mid Y=k) * P(Y=k)}{P(X1) * P(X2) \ldots * P(Xn)}$$

can be understood as ..

$$\text{Probability of Outcome I Evidence (Posterior Probability)} = \frac{\text{Probability of Likelihood of evidence} * \text{Prior}}{\text{Probability of Evidence}}$$

Probability of Evidence is same for all classes of Y

Naive Bayes Example by Hand

- Say you have 1000 fruits which could be either ‘banana’, ‘orange’ or ‘other’.
- These are the 3 possible classes of the Y variable.
- We have data for the following X variables, all of which are binary (1 or 0).
- Long
- Sweet
- Yellow

The first few rows of the training dataset look like this

Fruit	Long (x1)	Sweet (x2)	Yellow (x3)
Orange	0	1	0
Banana	1	0	1
Banana	1	1	1
Other	1	1	0
..	..	..	..

For the sake of computing the probabilities, let's aggregate the training data to form a counts table like this.

Type	Long	Not Long	Sweet	Not Sweet	Yellow	Not Yellow	**Total**
Banana	400	100	350	150	450	50	500
Orange	0	300	150	150	300	0	300
Other	100	100	150	50	50	150	200
Total	500	500	650	350	800	200	1000

Objective

- So the objective of the classifier is to predict if a given fruit is a ‘Banana’ or ‘Orange’ or ‘Other’ when only the 3 features (long, sweet and yellow) are known.
- Let’s say you are given a fruit that is: Long, Sweet and Yellow, can you predict what fruit it is?
- This is the same of predicting the Y when only the X variables in testing data are known. Let’s solve it by hand using Naive Bayes.
- The idea is to compute the 3 probabilities, that is the probability of the fruit being a banana, orange or other. Whichever fruit type gets the highest probability wins.

Step 1: Compute the 'Prior' probabilities for each of the class of fruits.

- That is, the proportion of each fruit class out of all the fruits from the population. You can provide the 'Priors' from prior information about the population. Otherwise, it can be computed from the training data.
- For this case, let's compute from the training data. Out of 1000 records in training data, you have 500 Bananas, 300 Oranges and 200 Others. So the respective priors are 0.5, 0.3 and 0.2.
- P(Y=Banana) = 500 / 1000 = 0.50
- P(Y=Orange) = 300 / 1000 = 0.30
- P(Y=Other) = 200 / 1000 = 0.20

Step 2: Compute the probability of evidence that goes in the denominator.

- This is nothing but the product of P of Xs for all X. This is an optional step because the denominator is the same for all the classes and so will not affect the probabilities.
- P(x1=Long) = 500 / 1000 = 0.50
- P(x2=Sweet) = 650 / 1000 = 0.65
- P(x3=Yellow) = 800 / 1000 = 0.80

Step 3: Compute the probability of likelihood of evidences that goes in the numerator.

- It is the product of conditional probabilities of the 3 features. If you refer back to the formula, it says P(X1 |Y=k). Here X1 is ‘Long’ and k is ‘Banana’. That means the probability the fruit is ‘Long’ given that it is a Banana. In the above table, you have 500 Bananas. Out of that 400 is long. So, P(Long | Banana) = 400/500 = 0.8.
- Here, I have done it for Banana alone.
- **Probability of Likelihood for Banana**
- P(x1=Long | Y=Banana) = 400 / 500 = 0.80
- P(x2=Sweet | Y=Banana) = 350 / 500 = 0.70
- P(x3=Yellow | Y=Banana) = 450 / 500 = 0.90
- So, the overall probability of Likelihood of evidence for Banana = 0.8 * 0.7 * 0.9 = 0.504

Step 4: Substitute all the 3 equations into the Naive Bayes formula, to get the probability that it is a banana.

Step 4: If a fruit is *'Long', 'Sweet' and 'Yellow'*, what fruit is it?

$$P(\text{Banana} \mid \textit{Long, Sweet and Yellow}) = \frac{P(\text{Long} \mid \text{Banana}) * P(\text{Sweet} \mid \text{Banana}) * P(\text{Yellow} \mid \text{Banana}) \times P(\text{banana})}{P(\text{Long}) * P(\text{Sweet}) * P(\text{Yellow})}$$

$$= \frac{0.8 * 0.7 * 0.9 * 0.5}{P(\text{Evidence})} = 0.252/P(\text{Evidence})$$

$$P(\text{Orange} \mid \textit{Long, Sweet and Yellow}) = 0, \text{ because } P(\text{Long} \mid \text{Orange}) = 0$$

$$P(\text{Other Fruit} \mid \textit{Long, Sweet and Yellow}) = 0.01875 / P(\text{Evidence})$$

Answer: Banana - Since it has highest probability amongst the 3 classes

Similarly, you can compute the probabilities for 'Orange' and 'Other fruit'. The denominator is the same for all 3 cases, so it's optional to compute.

Clearly, Banana gets the highest probability, so that will be our predicted class.

What is Laplace Correction?

The value of P(Orange | Long, Sweet and Yellow) was zero in the above example, because, P(Long | Orange) was zero. That is, there were no ‘Long’ oranges in the training data.

It makes sense, but 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’. Most Naive Bayes model implementations accept this or an equivalent form of correction as a parameter.

What is Gaussian Naive Bayes?

- So far 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 you assume the X's follow a Normal (aka Gaussian) Distribution, which is fairly common, we substitute the corresponding probability density of a Normal distribution and call it the Gaussian Naive Bayes. You need just the mean and variance of the X to compute this formula.

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

where mu and sigma are the mean and variance of the continuous X computed for a given class 'c' (of Y).

To make the features more Gaussian like, you might consider transforming the variable using something like the Box-Cox to achieve this.

Lets code:

- https://www.machinelearningplus.com/predictive-modeling/how-naive-bayes-algorithm-works-with-example-and-full-code/