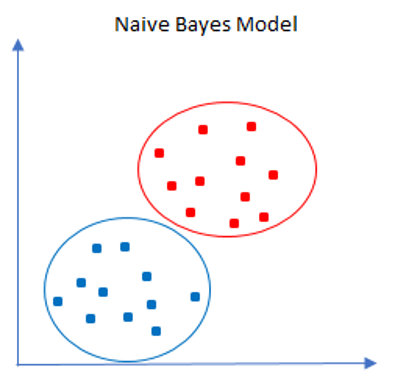
Naive Bayes Classifier

**What is a classifier?**

A classifier is a machine learning model that is used to discriminate different objects based on certain features.

**Principle of Naive Bayes Classifier:**

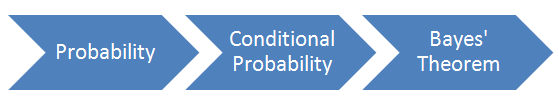
A Naive Bayes classifier is a probabilistic machine learning model that’s used for classification task. The crux of the classifier is based on the Bayes theorem. **Naïve Bayes** is a supervised learning algorithm used for classification tasks. Hence, it is also called Naive Bayes Classifier.



As other supervised learning algorithms, **Naïve Bayes** uses features to make a prediction on a target variable. The key difference is that **Naïve Bayes** assumes that features are independent of each other and there is no correlation between features. However, this is not the case in real life. This naive assumption of features being uncorrelated is the reason why this algorithm is called “naive”.

**Probability and conditional probability**

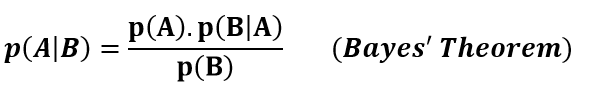
How about the “bayes”? Bayes comes from the famous [Bayes’ Theorem](https://en.wikipedia.org/wiki/Bayes%27_theorem) of Thomas Bayes. To get a comprehensive understanding of Bayes’ Theorem, we should talk about probability and conditional probability first.



Probability simply means the likelihood of an event to occur and always takes a value between 0 and 1 (0 and 1 inclusive). The probability of event A is denoted as **p (A)** and calculated as the number of the desired outcome divided by the number of all outcomes. For example, when you roll a die, the probability of getting a number less than three is 2 / 6. The number of desired outcomes is 2 (1 and 2); the number of total outcomes is 6.

Conditional probability is the likelihood of an event A to occur given that another event that has a relation with event A has already occurred. Suppose that we have 6 blue balls and 4 yellows placed in two boxes as seen below. I ask you to randomly pick a ball. The probability of getting a blue ball is 6 / 10 = 0,6. What if I ask you to pick a ball from box A? The probability of picking a blue ball clearly decreases. The condition here is to pick from box A which clearly changes the probability of the event (picking a blue ball). The probability of event A given that event B has occurred is denoted as **p (A|B)**.

**Bayes Theorem:**

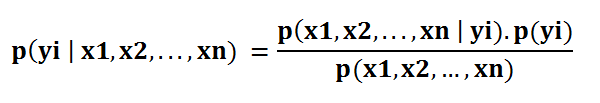


Now we have an understanding of Bayes’ Theorem. It’s time to see how **Naïve Bayes** classifier uses this theorem.

Using Bayes theorem, we can find the probability of **A** happening, given that **B** has occurred. Here, **B** is the evidence and **A** is the hypothesis. The assumption made here is that the predictors/features are independent. That is presence of one particular feature does not affect the other. Hence it is called naive.

**Naïve Bayes Classifier**

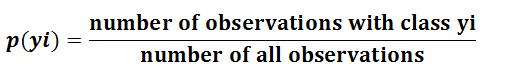
**Naïve Bayes** is a supervised learning algorithm for classification so the task is to find the class of observation (data point) given the values of features. **Naïve Bayes** classifier calculates the probability of a class a set of feature values (i.e. p(yi | x1, x2 , … , xn)). Input this into Bayes’ theorem:



**p(x1, x2 , … , xn | yi)**means the probability of a specific combination of features given a class label. To be able to calculate this, we need extremely large datasets to have an estimate on the probability distribution for all different combinations of feature values. To overcome this issue, **Naïve Bayes** **algorithm assumes that all features are independent of each other.**

Furthermore, denominator (p(x1,x2, … , xn)) can be removed to simplify the equation because it only normalizes the value of conditional probability of a class given an observation ( p(yi | x1,x2, … , xn)).

The probability of a class ( p(yi) ) is very simple to calculate:



Under the assumption of features being independent, **p(x1, x2 , … , xn | yi)**can be written as**:**

https://miro.medium.com/max/679/1*qIqZ-6m7GCts4XLesxwRLA.png

The conditional probability for a single feature given the class label (i.e. p(x1 | yi) ) can be more easily estimated from the data. The algorithm needs to store probability distributions of features for each class independently. For example, if there are 5 classes and 10 features, 50 different probability distributions need to be stored. Type of distributions depend on the characteristics of features:

* For binary features (Y/N, True/False, 0/1): Bernoulli distribution
* For discrete features (i.e. word counts): Multinomial distribution
* For continuous features: Gaussian (Normal) distribution

It is common to name the **Naïve Bayes** with the distribution of features (i.e. Gaussian naive bayes classifier). For mixed type datasets, a different type of distribution may be required for different features.

Adding all these up, it became an easy task for Naïve Bayes algorithm to calculate the probability to observe a class given values of features (**p (yi | x1, x2, xn)).**

**Types of Naïve Bayes** **Classifier:**

**Multinomial Naïve Bayes**

This is mostly used for document classification problem, i.e. whether a document belongs to the category of sports, politics, technology etc. The features/predictors used by the classifier are the frequency of the words present in the document.

**Bernoulli Naive Bayes:**

This is similar to the multinomial **Naïve Bayes** but the predictors are Boolean variables. The parameters that we use to predict the class variable take up only values yes or no, for example if a word occurs in the text or not.

**Gaussian Naive Bayes:**

Perhaps the easiest Naïve Bayes classifier to understand is Gaussian **Naïve Bayes** Classifier.When the predictors take up a continuous value and are not discrete; we assume that these values are sampled from a Gaussian distribution.

**Pros and Cons of Naive Bayes Algorithm**

**Pros:**

* The assumption that all features are independent makes Naïve Bayes algorithm **very fast**compared to complicated algorithms.In some cases, speed is preferred over higher accuracy.
* It works well with high-dimensional data such as text classification, email spam detection.
* It is not only a simple approach but also a fast and accurate method for prediction.
* Naive Bayes has very low computation cost.
* It is easy and fast to predict class of test data set. It also perform well in multi class prediction
* When assumption of independence holds, a Naive Bayes classifier performs better compare to other models like logistic regression and you need less training data.
* It performs well in case of discrete response variable compared to the continuous variable.
* It also performs well in the case of text analytics problems.
* When the assumption of independence holds, a Naive Bayes classifier performs better compared to other models like logistic regression.

**Cons:**

* If categorical variable has a category in test data set, which was not observed in training data set, then model will assign a zero probability and will be unable to make a prediction. This is often known as “Zero Frequency”.
* Naive Bayes is also known as a bad estimator, so the probability outputs from predict\_proba are not to be taken too seriously.
* Another limitation of Naive Bayes is the assumption of independent predictors. In real life, it is almost impossible that we get a set of predictors which are completely independent.
* The assumption that all features are independent is not usually the case in real life so it makes Naïve-Bayes algorithm less accurate than complicated algorithms. Speed comes at a cost.

**Conclusion**

They are fast and easy to implement but their biggest disadvantage is that the requirement of predictors to be independent. In most of the real life cases, the predictors are dependent; this hinders the performance of the classifier.

**Real time Prediction:** Naive Bayes is an eager learning classifier and it is sure fast. Thus, it could be used for making predictions in real time.

**Multi class Prediction:** This algorithm is also well known for multi class prediction feature.

**Text classification/ Spam Filtering/ Sentiment Analysis:** Naive Bayes classifiers mostly used in text classification (due to better result in multi class problems and independence rule) have higher success rate as compared to other algorithms.