The picture can't be displayed.

Machine Learning Project

Supervised by :

Pr.ABDELHAK MAHMOUDI

Realized by :

FATIMA ZAHRA EL HAJJI

جامعة محمد الخامس بالرباط
Université Mohammed V de Rabat

ENSIAS

Plan :

1. Random Forest Algorithm
2. Gaussian Mixture Algorithm

Random Forest

“The majority of which each member taken apart is not a remarkable man, is however above the superior men “ ,Aristotle-Politics

Random Forest

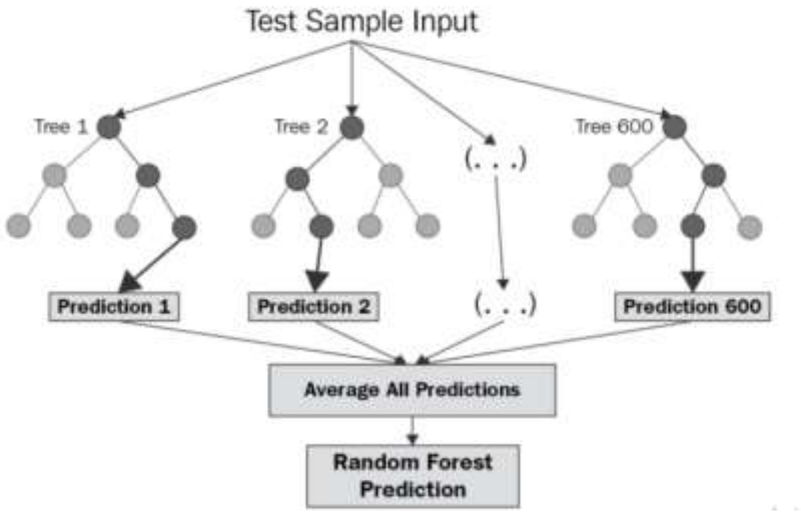
In Random Forest the first word that attracts us is “**Forest**”. We therefore understand that this algorithm will be based on trees that we call a **decision trees**. And for the second word "**Random**", we understand that the construction of these trees will be random.

Decision trees and their problems :

Decision tree is a supervised algorithm, used to predict a value (prediction) or a category (classification). As its name suggests, this algorithm is based on the construction of a tree which makes the method quite simple to explain and easier to interpret. But there are some limitations to this algorithm :

- **Overfitting:** this problem occurs when we focus on trees that are too complex, too dependent on the sample used for training.
- **Variance error:** Decision trees have high variance, which means that small changes in the learning data can lead to large changes in the end result.

To remedy the weaknesses of decision trees, we turn to **Random Forest** which illustrates the power of combining several decision trees into a single model.

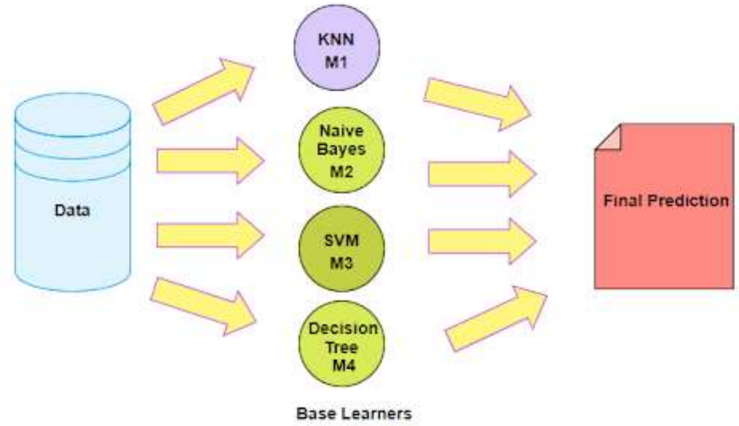


Random Forest Structure

Thus Random forest allows to use several decision trees to create a forest and improve ensemble generalization using **Ensemble Method** to make these trees work together.

Ensemble Method :

An Ensemble method is a technique that combines the predictions from multiple machine learning algorithms together to make more accurate predictions than any individual model. A model comprised of many models is called an Ensemble model,the ensemble model tends to be more flexible (less bias) and less data-sensitive.



Ensemble Learning Method

There are two types of Ensemble Method :

- **Bagging :** Training a bunch of individual models in a parallel way. Each model is trained by a random subset of the data
- **Boosting :** Training a bunch of individual models in a sequential way. Each individual model learns from mistakes made by the previous model.

How the Random Forest Algorithm Works ?

The following are the basic steps involved in performing the random forest algorithm:

1. we pick N random records from the dataset.
2. we build a decision tree based on these N records.
3. we choose the number of trees you want in your algorithm and repeat steps 1 and 2.
4. In case of a **regression problem**, for a new record, each tree in the forest predicts a value for Y (output). The final value can be calculated by taking the average of all the values predicted by all the trees in forest. Or, in case of a **classification problem**, each tree in the forest predicts the category to which the new record belongs. Finally, the new record is assigned to the category that wins the majority vote.

Advantages of Random Forest :

- It is one of the most accurate learning algorithms available. For many data sets, it produces a highly accurate classifier.
- It runs efficiently on large databases.
- It can handle thousands of input variables without variable deletion.
- It has an effective method for estimating missing data and maintains accuracy when a large proportion of the data are missing.

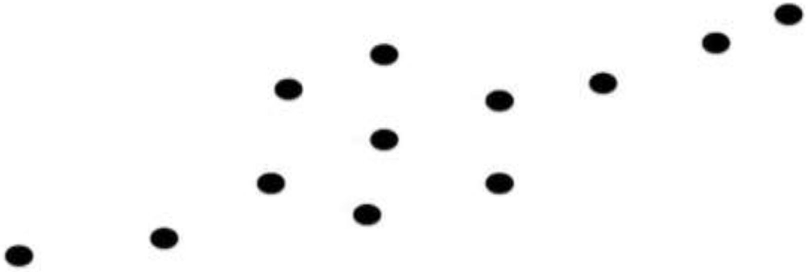
Disadvantages of Random Forest :

- Model interpretability: Random forest models are not all that interpretable; they are like black boxes.
- For very large data sets, the size of the trees can take up a lot of memory.

Gaussian Mixture

Gaussian Mixture

Sometimes datasets are complicated like this. In this dataset it looks like there are two clusters and these clusters seem to intersect which makes the problem very difficult for traditional clustering algorithms like k-means.



So we need something different

One important characteristic of **K-means** is that it is a **hard clustering method**, which means that it will associate each point to one and only one cluster. A limitation to this approach is that there is no uncertainty measure or **probability** that tells us how much a data point is associated with a specific cluster.

So, can we move from **hard clustering** to **soft clustering** to improve the representation of our clusters?

This is exactly what **Gaussian mixing** models, or simply MGMs, attempt to do.

In Gaussian Mixture the first word that attracts us is “**Gaussian**” , which means that the algorithm is based on the normal or Gaussian distribution of the samples. And for the second word "**Mixture**", means that there is the superposition of several Gaussian distributions.

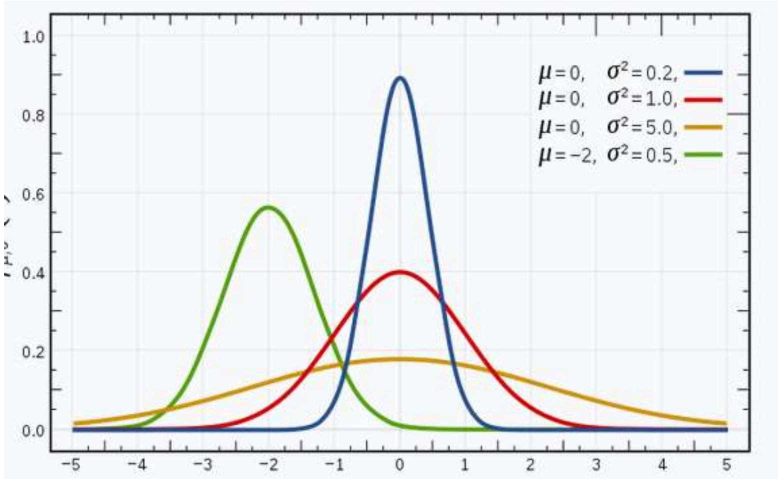
Broadly, Gaussian mixture models are **probabilistic models** and use the **soft clustering** approach to distribute the points in different clusters :

- **Soft clustering** means that a sample point can belong to several clusters which is not the case in the k-means algorithm where each one belongs to a particular cluster.
- **Probabilistic model** model means that the model indicates the probability that a data point is associated with a specific cluster.

The Gaussian Distribution :

The Gaussian distribution, normal distribution, is a probability distribution which accurately models a large number of phenomena in the world.It has a bell-shaped curve, with the data points symmetrically distributed around the mean value.

This image shows some Gaussian distributions with a difference in mean (μ) and variance (σ2). Remember that the higher the σ value, the higher the spread.

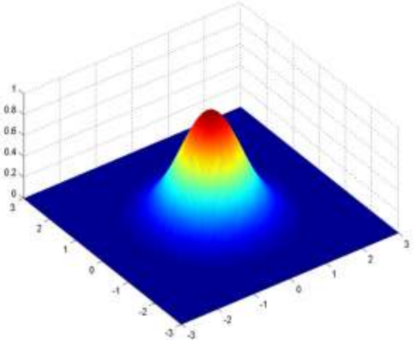


In a one dimensional space, the probability density function of a Gaussian distribution is given by:

$$f(x \mid \mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

where μ is the mean and σ2 is the variance.

But this would only be true for a single variable. In the case of two variables, instead of a 2D bell-shaped curve, we will have a 3D bell curve.



The probability density function would be given by:

$$f(x \mid \mu, \Sigma) = \frac{1}{\sqrt{2\pi|\Sigma|}} \exp\left[-\tfrac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})'\Sigma^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\right]$$

where x is the input vector, μ is the 2D mean vector, and Σ is the 2×2 covariance matrix. The covariance would now define the shape of this curve. We can generalize the same for d-dimensions.

Thus, this multivariate Gaussian model would have x and μ as vectors of length d, and Σ would be a d x d covariance matrix.Hence, for a dataset with d features, we would have a mixture of k Gaussian distributions (where k is equivalent to the number of clusters), each having a certain mean vector and variance matrix.

But how are the mean value and the value of the variance for each Gaussian assigned?

These values are determined using a technique called **Expectation-Maximization** (EM).

Expectation-Maximization :

Expectation-Maximization (EM) is a statistical algorithm for finding the right model parameters. We typically use EM when the data has missing values, or in other words, when the data is incomplete.

Broadly, the Expectation-Maximization algorithm has two steps:

- E-step: In this step, the available data is used to estimate (guess) the values of the missing variables
- M-step: Based on the estimated values generated in the E-step, the complete data is used to update the parameters

So how does GMM use the concept of EM and how can we apply it for a given set of points?

Suppose we have two Gaussians A and B, then the EM (Expectation-Maximization) algorithm will allow us to find the parameters of these two distributions starting from random values and adjusting them as we go along until the likelihood of these models are maximum. The steps are as follows:

- Initialize two normal laws A and B by choosing random values for (μA / σA and μB / σB)
- For each value of X, calculate its probability under the hypothesis A (pA) then B (pB)
- For each value of X, calculate the weight wA = pA / (pA + pB) and wB = pB / (pA + pB)
- Calculate new parameters (μA, σA) and (μB, σB) by fitting X from the weights wA and wB.
- Restart ...

After this brief presentation of the two Random forest and Gaussian mixture algorithms, we will try in the two notebooks to apply the first algorithm on Heart Disease data and the second on Medical Cost Personal data