Generation of the dataset

General information

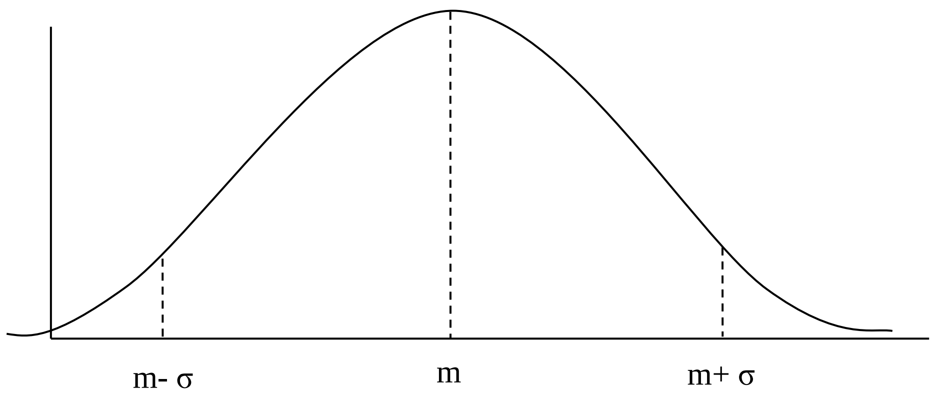
For a better analysis of the results of the learning algorithms that will be developed in this lab, the dataset that will be used will contain a lot of points that can be represented into a two-dimensional space. Thus, each input (sample) in the dataset will have two features: the first feature represents the *x*-coordinate and the second feature represent the *y*-coordinate of a point that will be drawn on the screen. In this idea I will propose an algorithm, based on Gauss function, that generates randomly these points.

Gaussian probability

The Gaussian probability is the probability that a point to be into the interval [*m*-*σ*, *m* + *σ*], where *m* is the center (middle range of curve) and *σ* is the dispersion of data.

The Gauss function:

$$Gauss(x) = e^{-\frac{(m-x)^2}{2\sigma^2}}$$



Random coordinate generation algorithm

This algorithm will be run independently for each feature of a sample from the dataset (for the *x*-coordinate and after that, for the *y*-coordinate of each point). The parameters *m* and *σ* are specified, as inputs of the algorithm, for each group of samples that will be created. Thus, each data group to be created will have *mx*, *my*, σ_x and σ_y own. Also, each group will draw using a different color. The values of *x* and *y* will be represented in Cartesian coordinates (-300,300) not screen coordinates and will be saved in a file, each point on a line, along with the group number.
The algorithm:

1. Randomly select a group to generate a coordinate.

2. Randomly choose a value for the *x*-coordinated, in the *x* domain, for example (-300,300);
3. Calculate the probability that the randomly selected value (in step 2) is or is not closer to the center *m* (using the Gauss function);
4. Randomly generate a probability in the range [0,1];
5. Check whether the probability computed for the selected coordinates (in step 3) is higher than the probability randomly selected (in step 4);
   a. If yes, the randomly selected coordinates (in step 2) are taken into account and proceed to step 6;
   b. If not, go back to step 2;
6. Perform steps 2-5 for the y- coordinated, then in step 5.a algorithm ends for a point (a sample).

For a given center *m* and dispersion σ, the algorithm will generate a random number of points around *m*.

For this lab it is necessary to make an application that generates exactly 10000 points for 5 different centers (specified by *mx*, *my*, σ_x and σ_y). The point will be written to a file, each point on a line, and also drawn on the screen. So, we can easily see that they are in the areas we specify. In the file the points will be written in the order of generation, so the generated points for a certain area will not be consecutive in the file.

Note:
The center of the workspace will be taken as the center of the coordinate axis (0,0). This means that the points will be in Cartesian coordinates (not Screen coordinates).