****

**STATISTICAL LEARNING**

**AND NEURAL NETWORKS**

**A.A. 2021/2022**

**Professor ENRICO MAGLI**

**Students FRANCESCO DONATO (s..)**

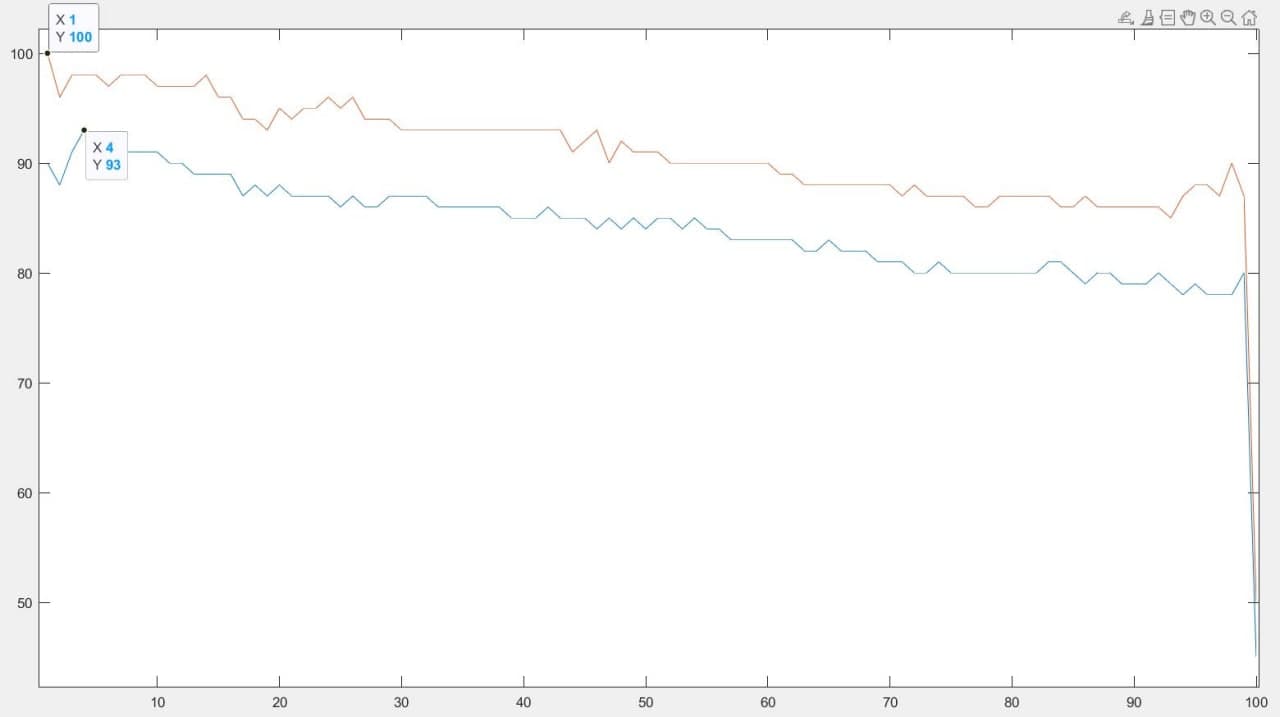
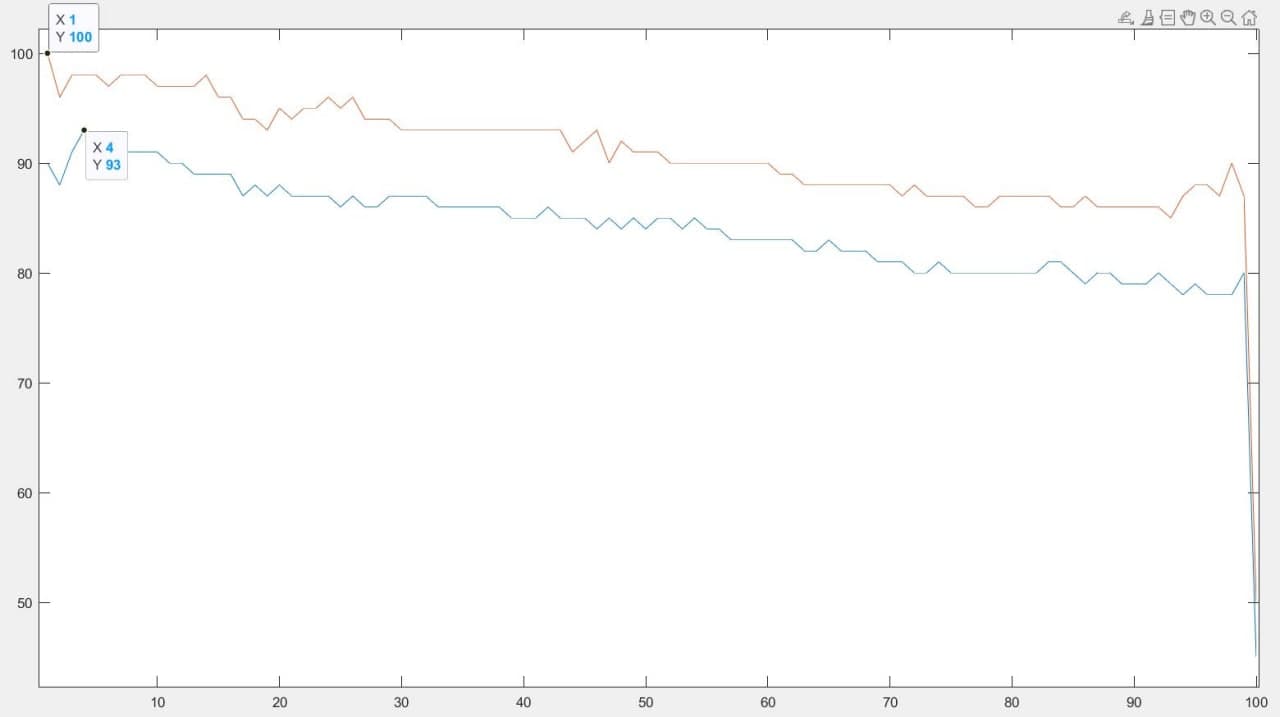
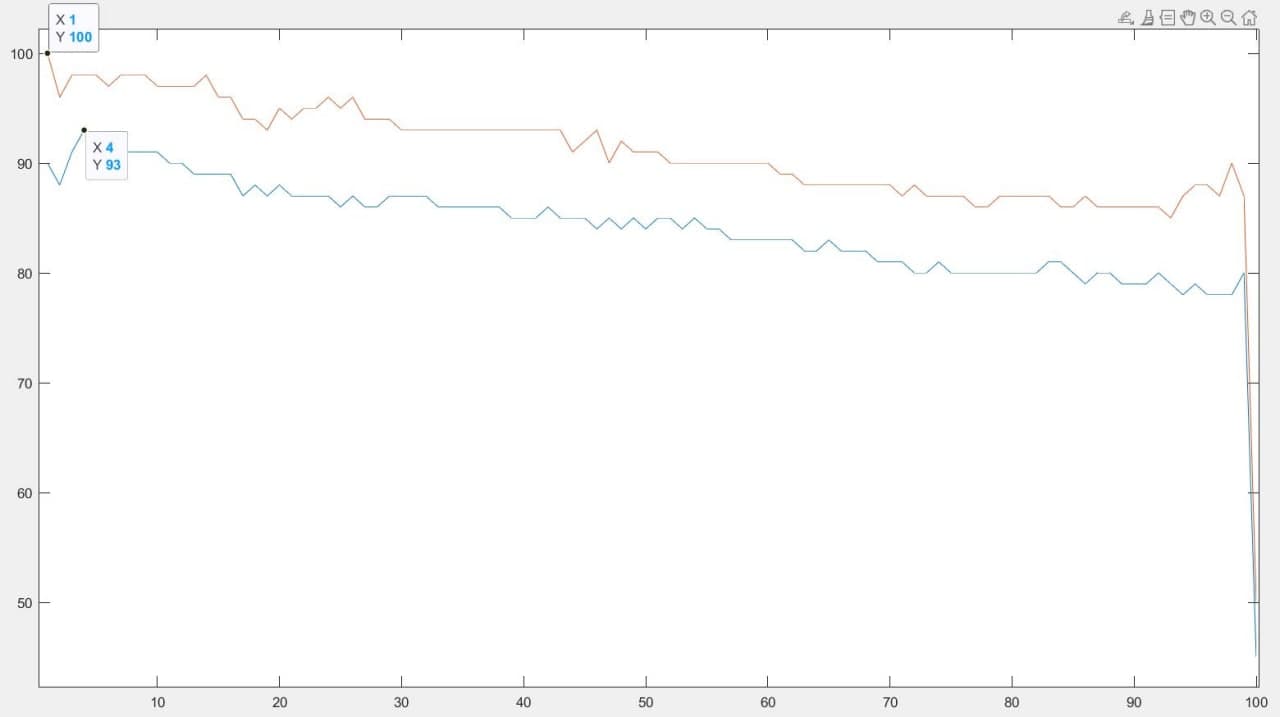
**MARCO COLOCRESE (s301227)**

**COMPUTER LAB1**

**Exercise 1**

In the first exercise we implemented a basic algorithm using the K-NN classifier.

For each K value (from 1 to 100) we ranked the vector containing all distances between the point we were analyzing and all the training set points. We did it for all test set points, hence we developed three nested ‘for’ loops. Having the sorted distance vector (regarding the current point), we counted the occurrences of each class among the nearest K points. The algorithm output was the class with the most occurrences, while in unbiased situations we let the algorithm choose a random value between the two classes.

The accuracy was evaluated computing the right/total ratio.

Test set accuracy

Training set accuracy

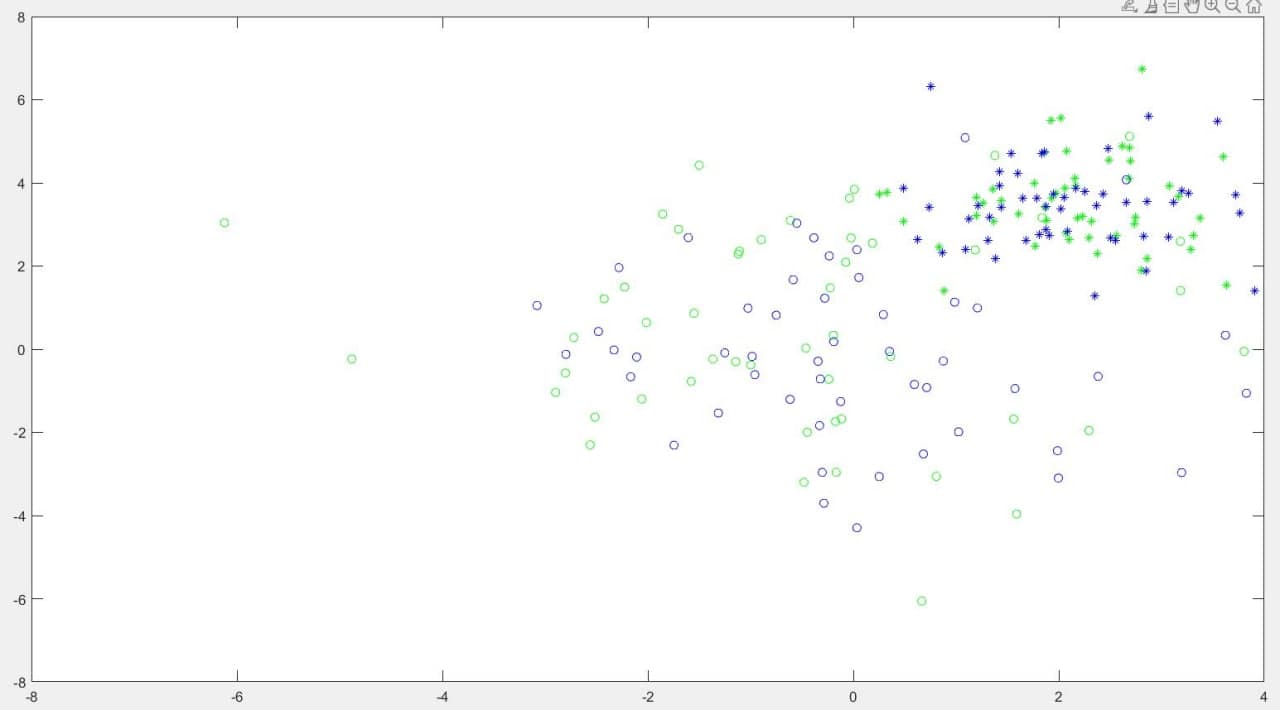
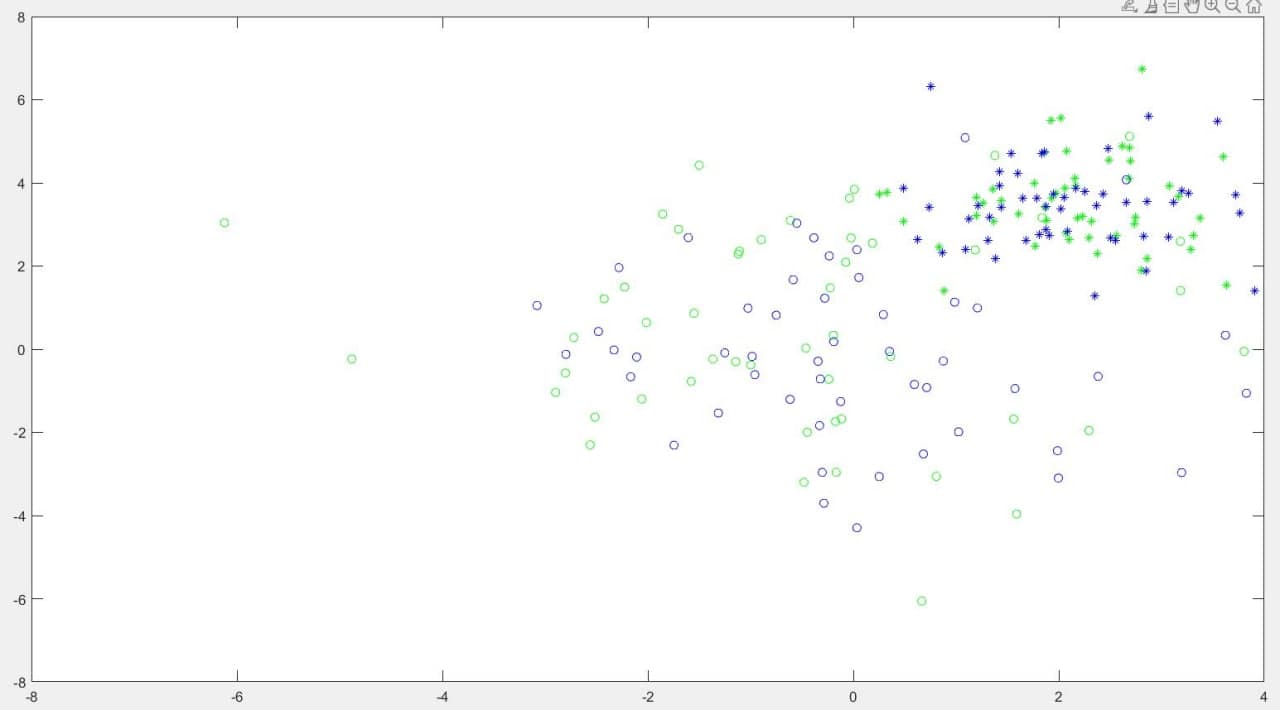
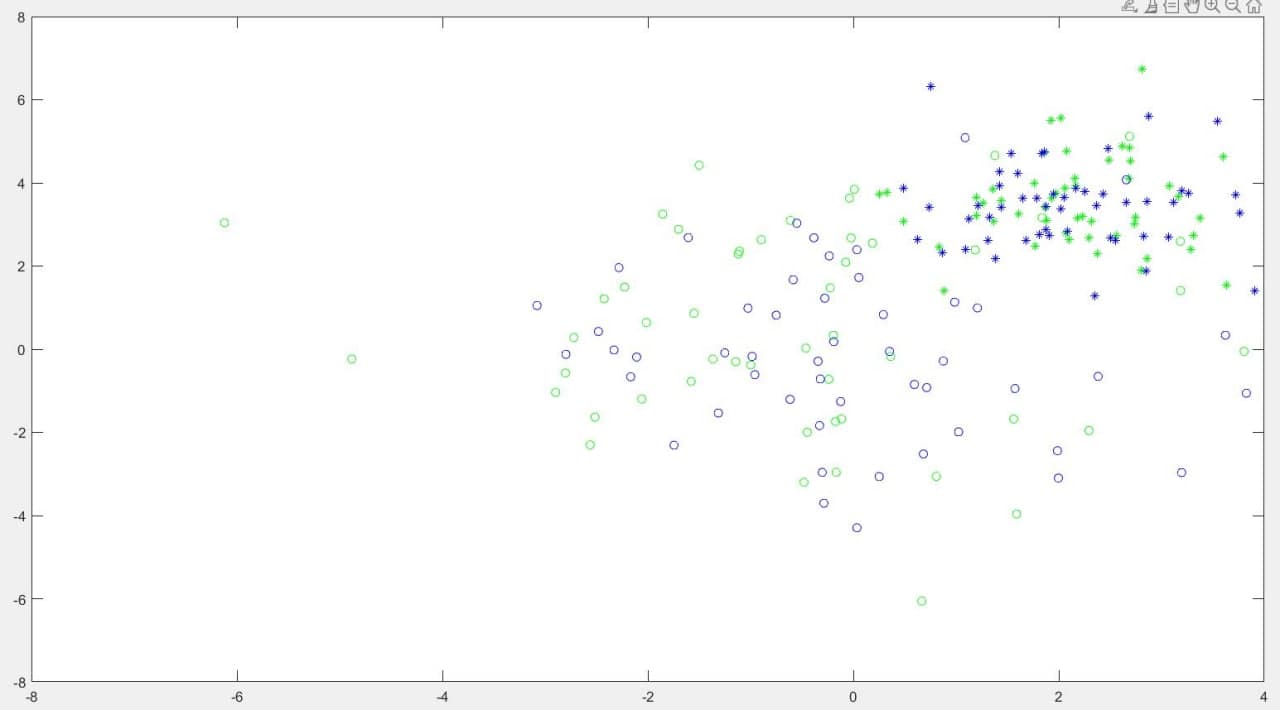
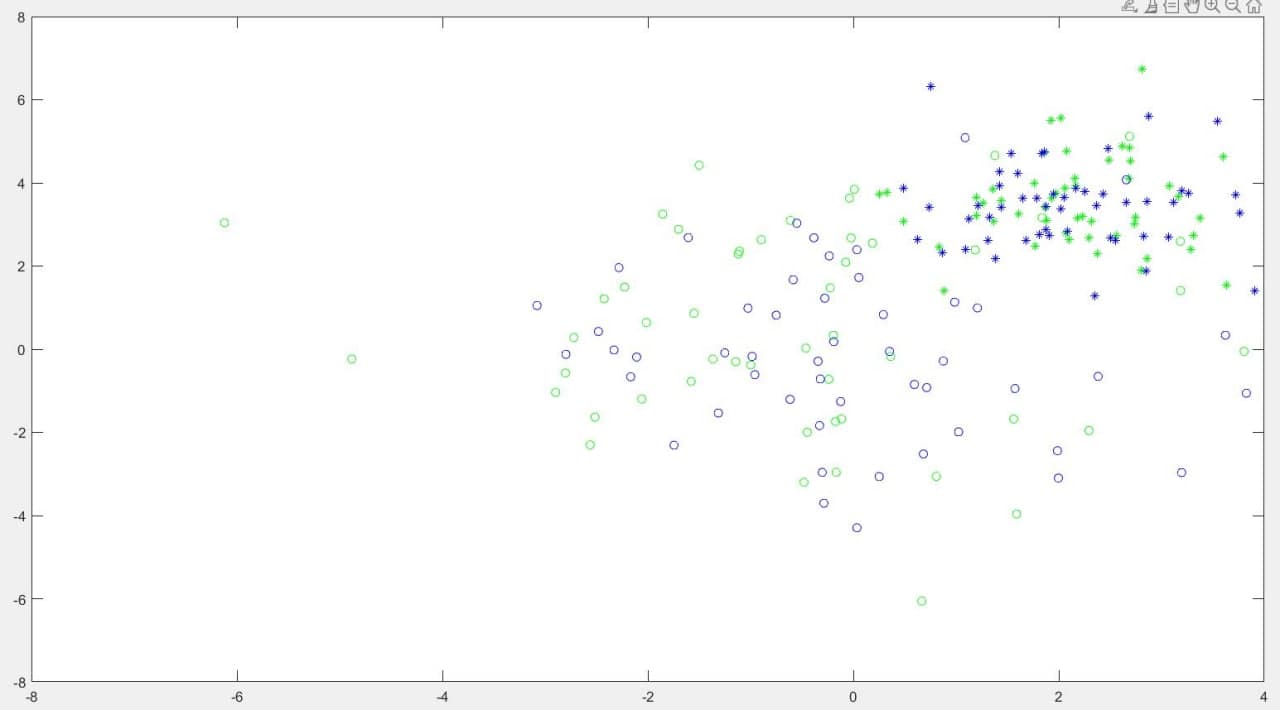
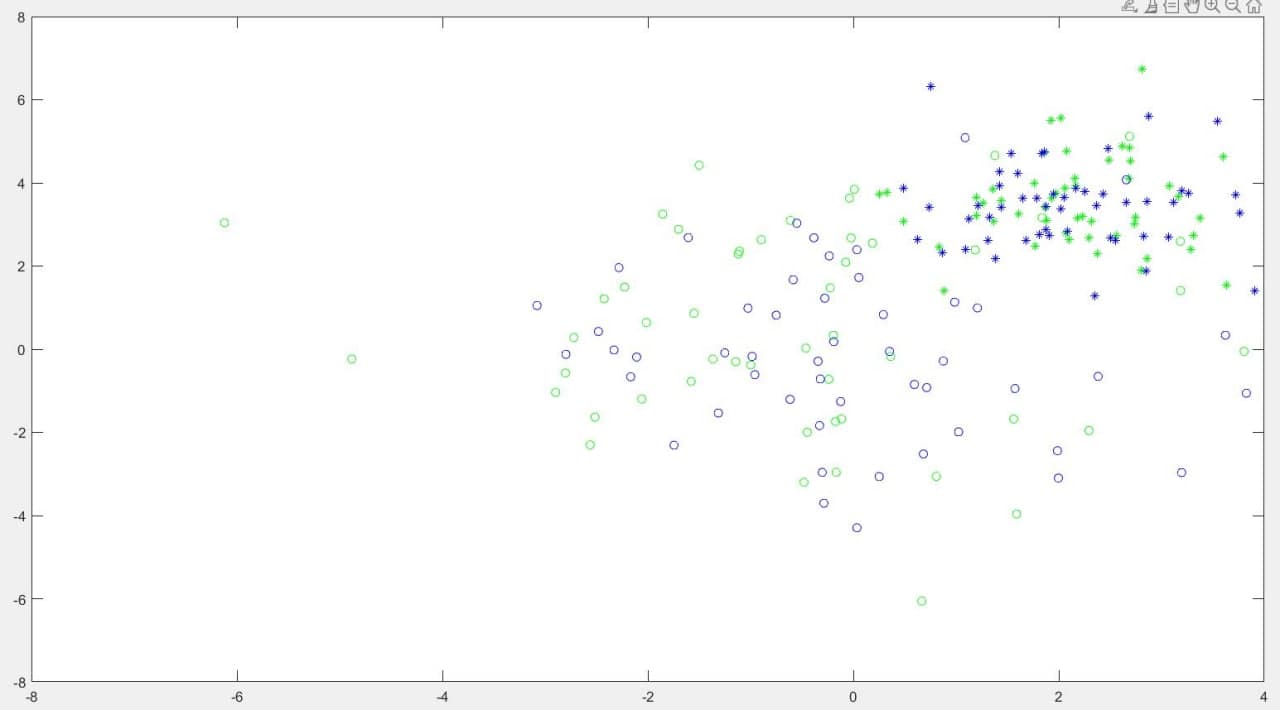
As predicted, for the training set values, the accuracy is overall higher, and in particular, results to be 100% for K=1.

Regarding this particular value we should observe the overfitting phenomenon in the Test set case.

In both cases, with the increasing of the K value, the accuracy tends to decrease significantly, falling to around 50% when K equals to 100. This accuracy value fluctuated around 50% as we repeatedly ran the program, because of our choice of assigning a random class to the point.

This decreasing shows the underfitting phenomenon, as with high values of K our algorithm is averaging too much.

Probably a good choice for K is a small value (4 or 5)



Test set – Class2

Training set – Class2

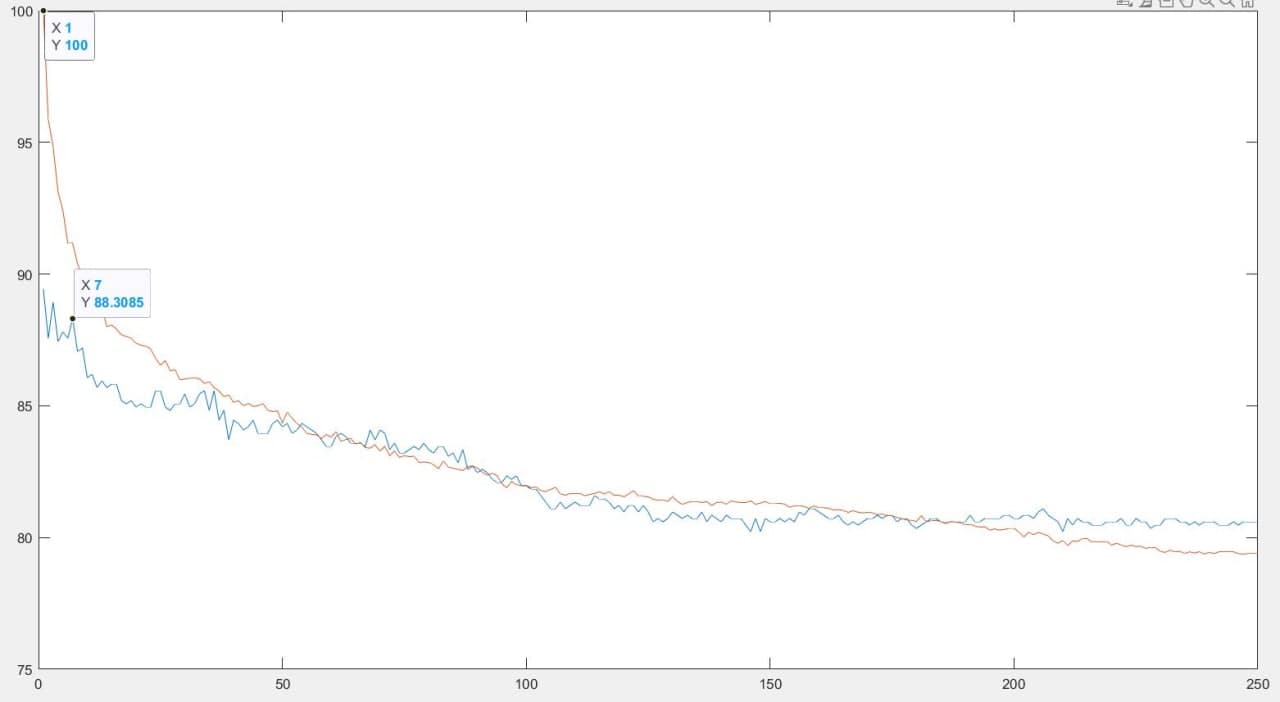
Test set – Class1

Training set – Class1

We can see that the error between the training set and the test set, regarding the accuracy, in the cases of overfitting and underfitting, is not significant. In our opinion, this derives from the distribution homogeneity of the points belonging to the two different classes. In other words, the clusters of data look almost isolated from each other (not mixed).

**Exercise 2**

This exercise is very similar to the first one, in fact we implemented the same algorithm, but with few modifications due to the different dataset. In particular, distances were calculated using 5 “coordinates”; having only the dataset, we divided it into training set and dataset, specifically 85% for the training set and 15% for the test set.



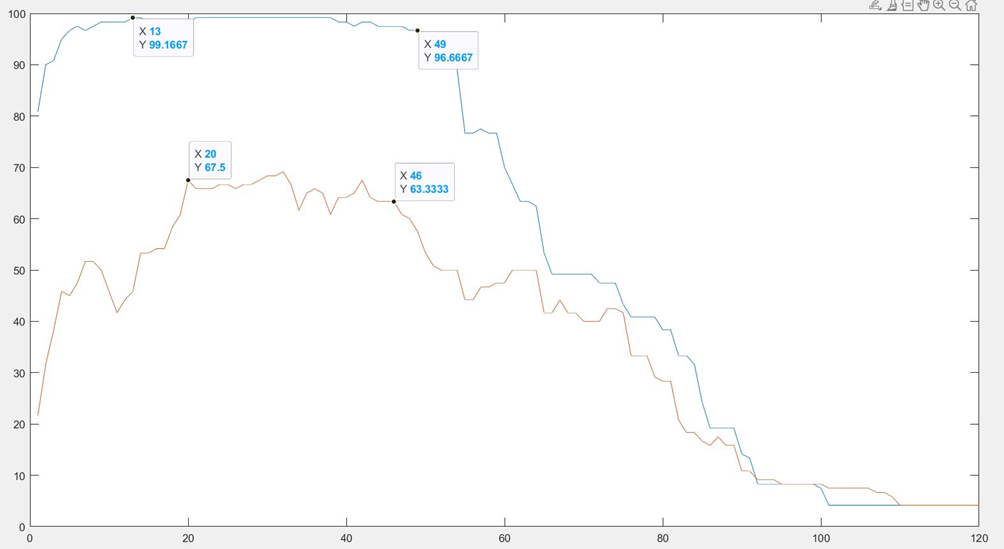
For high values of K we can observe the underfitting phenomenon, as the algorithm averages too much bringing to a decrease in terms of accuracy.

Also in this case, a good choice for K would be a small value (between 4 and 7).

**Exercise 3**

As in previous exercises, we applied the K-NN classifier to solve this classification problem. For each test set’s element we computed the “distances” (representing the absolute values of differences between the two RSSI values of the considered user and points belonging to the training set), we ordered the distance vectors and counted the occurrences of each class related to the K nearest points of each sensor. In other words, we considered K\*7 values for each user (K for each sensor). The classification output was obtained finding the class with most occurrences.

We implemented the algorithm with K going from 1 to 120 (because we had 120 points, hence the total number of measurements: 24\*5).



We noticed that when K=1, the accuracy (regarding the training set plot) doesn’t reach 100%. This happens because some measurements taken in different cells (classes) have the same value (ie. Train data (2,3,5) and Train data (2,3,6). Using the *max* function in Matlab, in case of equal values, the first one is taken. This implies that, among classes referring to distances between different points, the first one is taken as the output of our algorithm.

In the graph, overfitting and underfitting phenomenon are evident. When K is too small (left side of the graph), the complexity of the model implies overfitting), while on the right side the model is too simple because of the high K value, causing underfitting.

An optimal K vale would range from 20 to 40, where accuracy seems to mantain a good value.

**COMPUTER LAB1**

**Exercise 1**

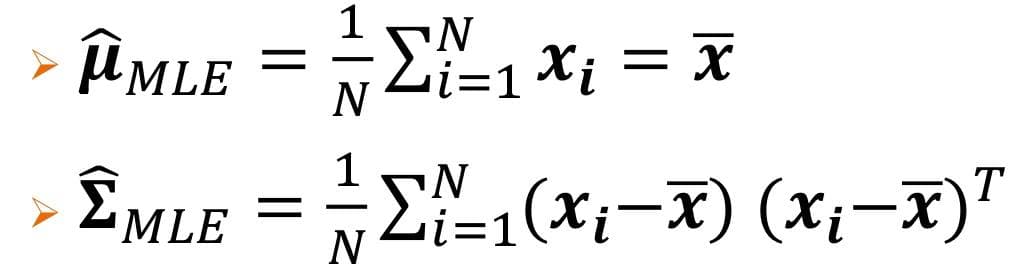
In the first exercise we used a dataset containing labelled data for two classes about height and weight of each person. We fitted a class-conditional Gaussian multivariate distribution to these data, and plotted the probability density functions.

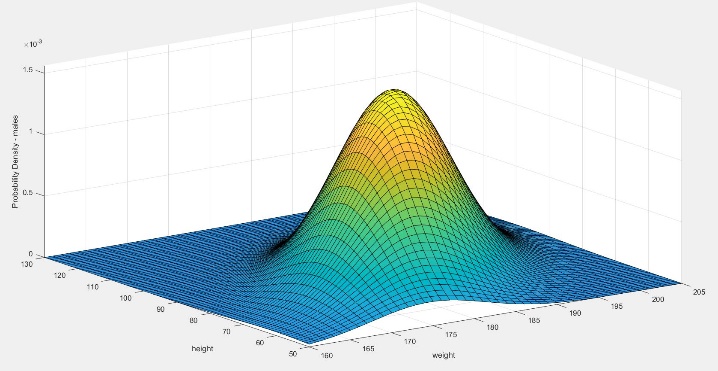
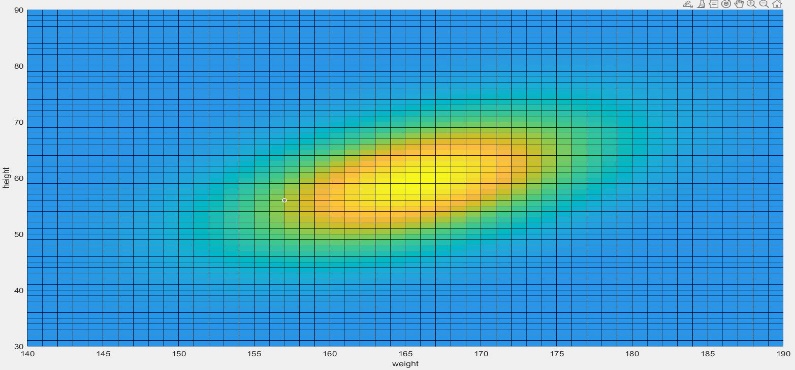
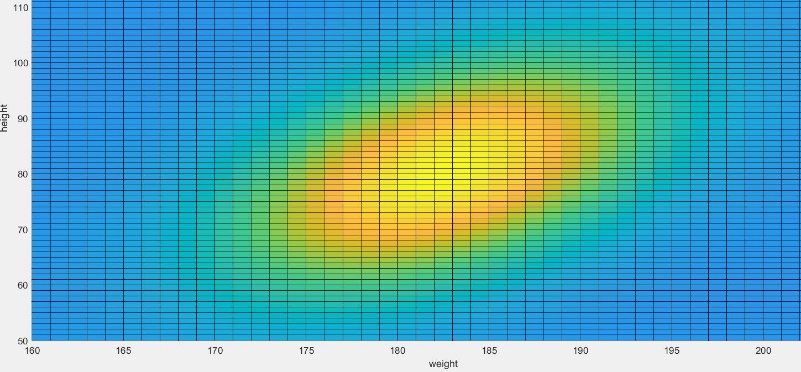
Through the scattered plots we guessed that a correlation between height and weight could exist (with the weight increasing, also the height tended to grow as well).

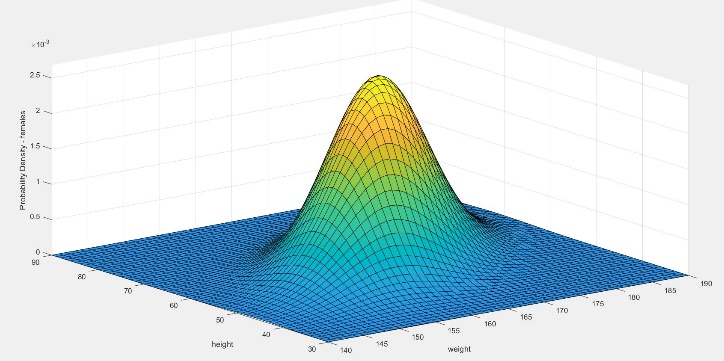
Visualizing the weight and height histograms, the distribution seemed to be gaussian, except for some outliers.

Knowing that combining gaussian distributions, we still obtain a gaussian one, we imagined that a gaussian model would be good enough for our data.

We calculated the maximum likelihood estimate of the mean and the covariance matrix, where each element of the mean vector is the average of the same feature (weight or height) of all males, and of all females.

xi is a two-element vector referring to the i-th person, containing the features. The mean vector x also has two elements (average weight and average height).





In the images on the left we can appreciate the multivariate gaussian distribution, where the peak is evident as the variance is relatively small. In the images on the right we can observe the elliptical contours which demonstrate the correlation between the two features, which are statistically dependent.

**Exercise 2**

In this exercise we fitted the parameters employed by a Naïve Bayes Classifier, using a Bernoulli model. To do this we had to calculate the prior probabilities of each class (, where c={1,2}) and the probability that the feature j was equal to one (meaning the presence of the word) in each class (, where we have 600 j features).

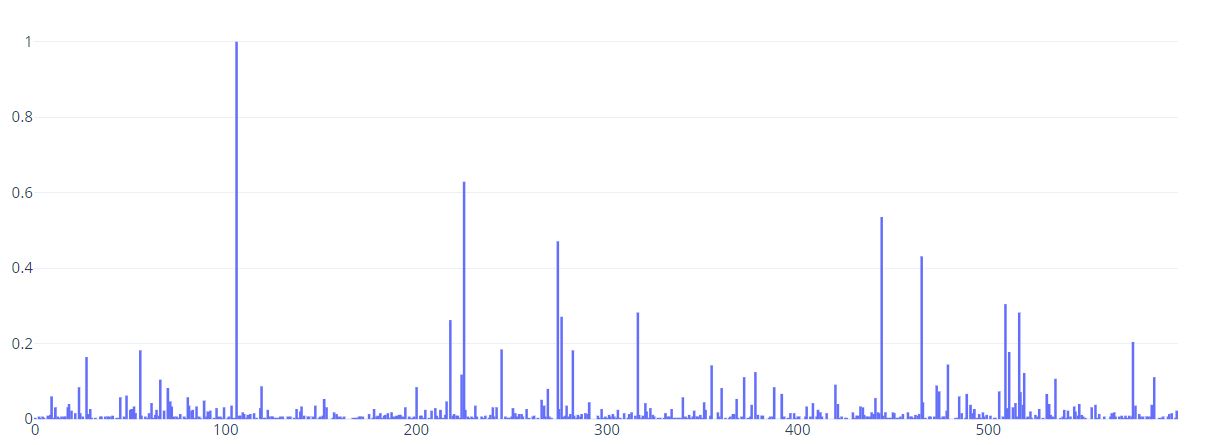


Immagine che contiene esterni, giorno

Descrizione generata automaticamenteclass 1

class 2

It can be seen that the trend is very similar between the two classes, infact setting an error of 10% of the mean (media) value of the probabilities, 79 features can be considered as uninformative ()