Introduction to machine learning
-
Project 1

DEFLANDRE Sophie - s160767

GÓMEZ HERRERA María Andrea Liliana - s198387

Master 2 Biomedical engineering
Master 1 Data Science

Academic year 2020-2021

1 Decision tree

1.1 Decision boundary in function of tree depth

a. Decision boundary illustration

The goal is to classify in which circle or ellipse a certain point is. Both data set are illustrated in figure 1 and 2.

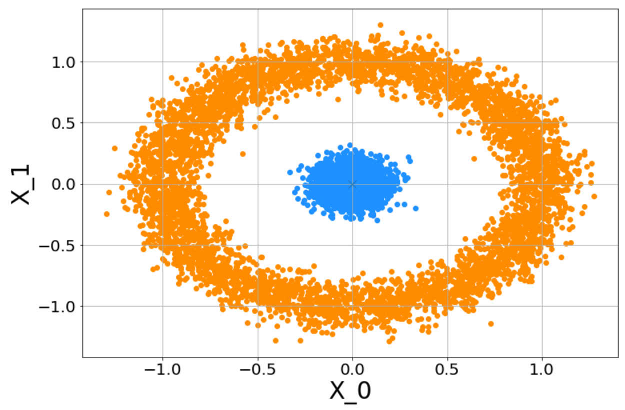


Figure 1: Data set 1

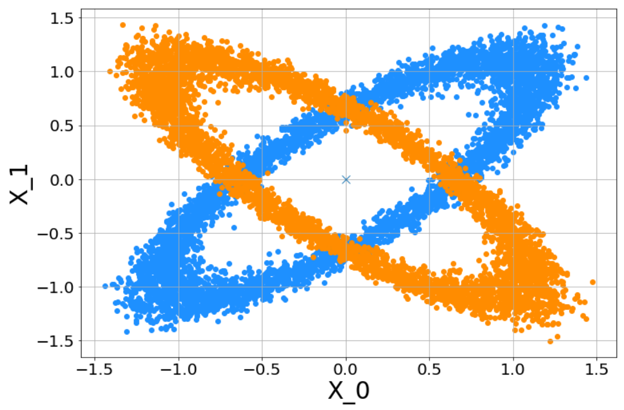
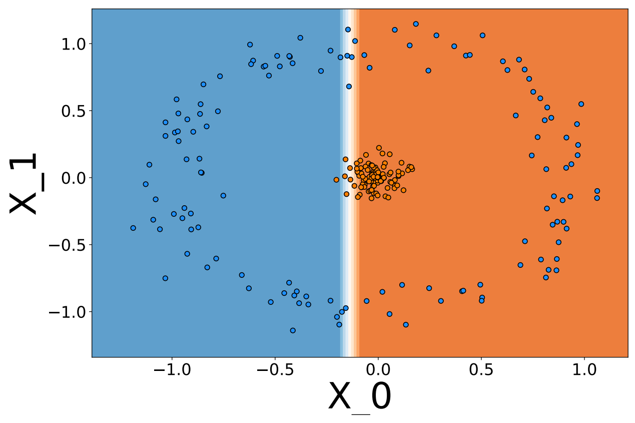


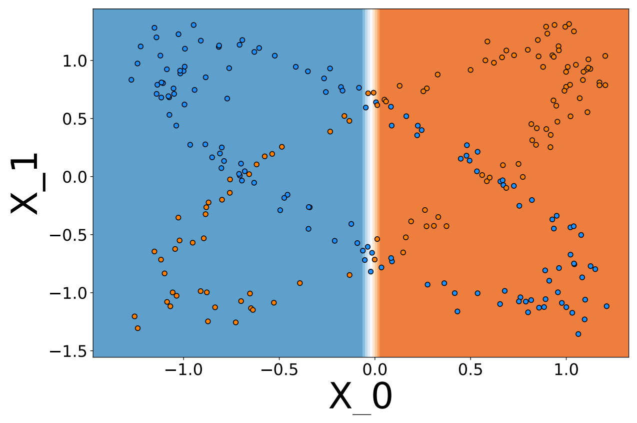
Figure 2: Data set 2

In order to achieve that, a tree model using the class DecisionTreeClassifier from scikitlearn is made. With this class we can define the maximum depth of the tree. The model is then trained with the learning sample. In figure 3 to 7, the decision boundary is represented for both models in function of the depth.

In figure 3 the depth is computed to 1 meaning that the data set is divided in two because only one decision is taken. By the way, for both problems the accuracy is very bad as the two classes are present in both halves of the plane.



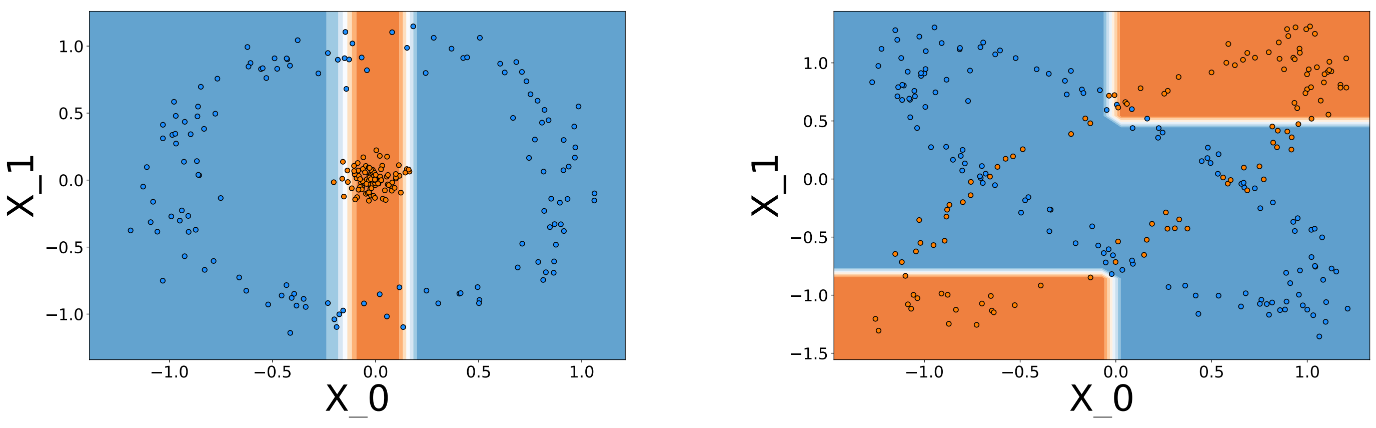
(a) Data set 1



(b) Data set 2

Figure 3: Tree with a depth of 1

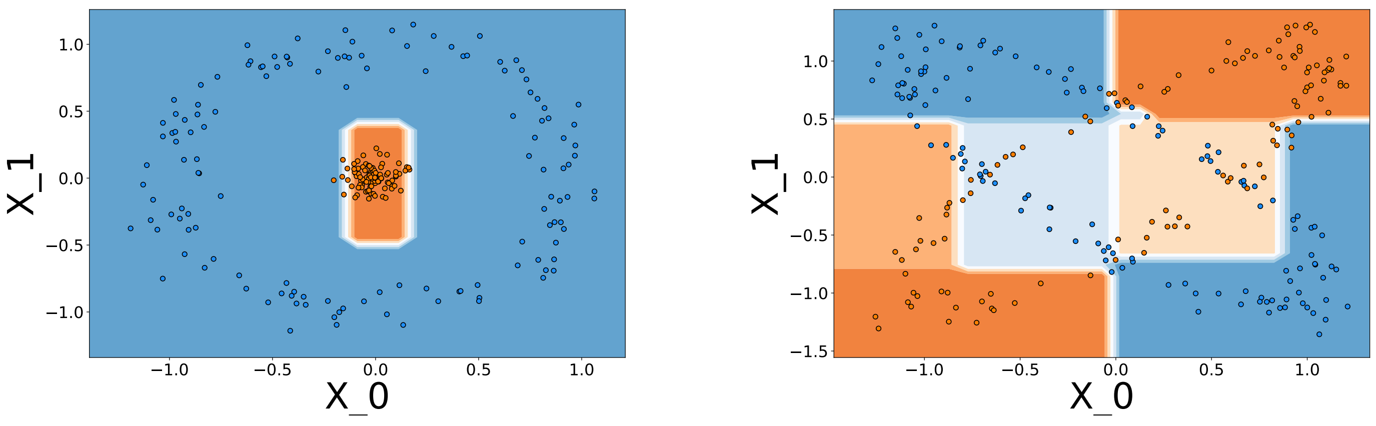
In figure 4, the trees depth is 2. The decision boundary is a bit better, one class of the data 1 is grouped correctly but there are elements corresponding to the other class in the same group, that implies that it does not provide a good classification.



(a) Data set 1 (b) Data set 2

Figure 4: Tree with a depth of 2

In figure 5, a depth of 4 is considered. For the first data set, the classification is already well done and no more improvement is needed. However, for the second data set the accuracy becomes better but stills not good yet due to miss classified data.

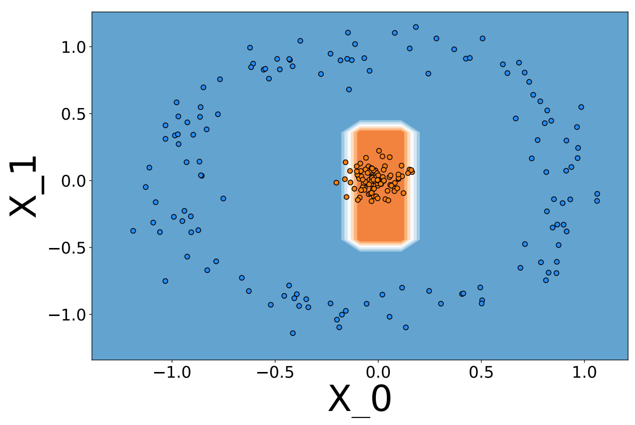


(a) Data set 1 (b) Data set 2

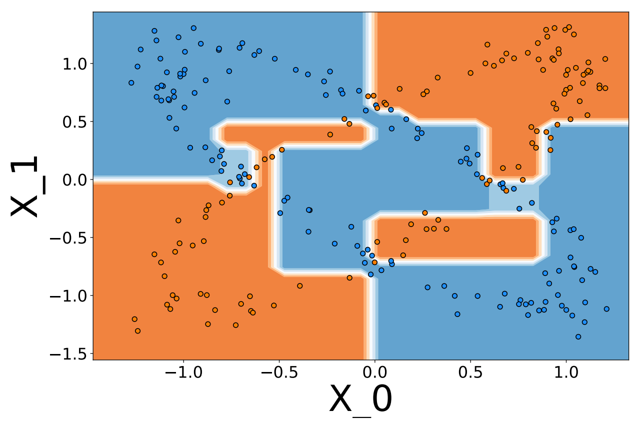
Figure 5: Tree with a depth of 4

In figure 6.a, we can see that the depth is still 4 because a depth of 8 would not improve the classification. In figure 6.b, we see that a depth of 8 improves the classification for data set 2. However, it becomes tricky because some parts contains both data from data set 1 and 2.

Finally, in figure 7 the depth is unconstrained. It does not change anything for data set 1 like the previous situation. The differences for the second data set are really small but the orange region is a bit bigger.
As a tree splits one variable at time we can observe that with all the depths the boundaries are aligned to the axis and rectangular regions are defined.

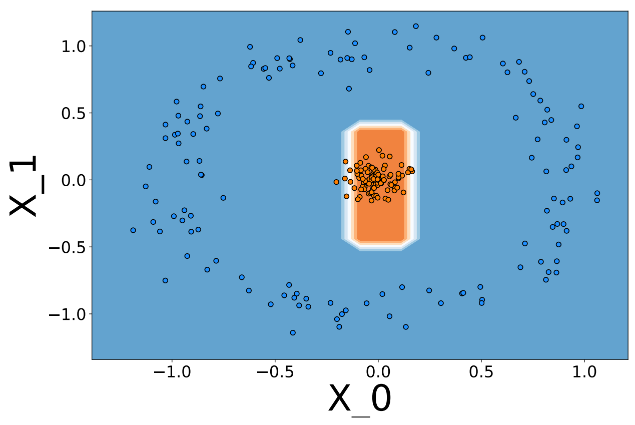


(a) Data set 1

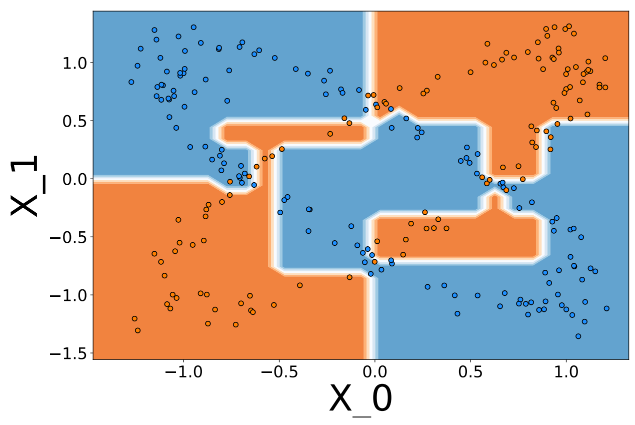


(b) Data set 2

Figure 6: Tree with a maximum depth of 8



(a) Data set 1



(b) Data set 2

Figure 7: Tree with a unconstrained depth

b. Under and over-fitting

For the first data set, the model is under-fitted when the depth is less than 4 because the orange region selects both types of observations. After reaching depth 4 there is no over-fitting neither for this first data set which is logical as the depth remains 4. This can be seen by plotting the learning and test error on the same figure like in figure 8.

The same way, the different values of the error are illustrated in figure 9 for the second data set. In this case, it seems to have an optimal depth with 8. Indeed, the error is higher than both learning and test sample with a lower depth (under-fitting) and the test error increases a little bit with a higher depth, we can observe over fitting with a depth of 12. Note that we mostly observe under-fitting because the size of the data set is small.

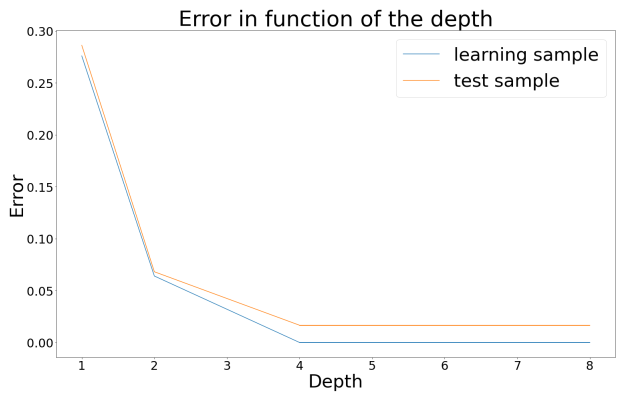


Figure 8: Data set 1

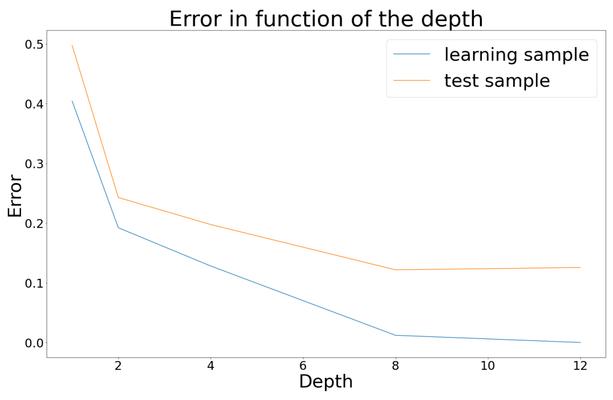
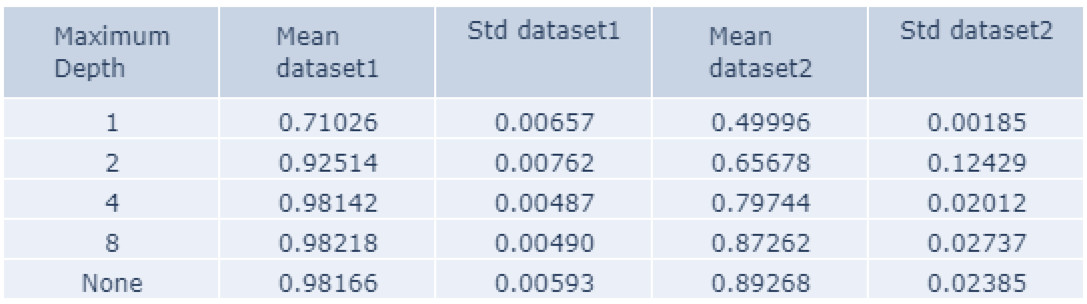


Figure 9: Data set 2

c. Explanation of unconstrained depth

The model seems more confident when the depth is unconstrained because the depth of the model will be chosen so that the classification is the best possible for the learning sample. However, in the case of the first data set, the model has a maximum depth of 4 so that not given a maximum depth will not change anything. For the second data set, a depth of 12 is chosen by the classifier and we observe a really good score for the learning sample but test sample score is a bit lower. The effect would be more pronounced for larger learning sample set.

1.2 Test set accuracies



Maximum Depth	Mean dataset1	Std dataset1	Mean dataset2	Std dataset2
1	0.71026	0.00657	0.49996	0.00185
2	0.92514	0.00762	0.65678	0.12429
4	0.98142	0.00487	0.79744	0.02012
8	0.98218	0.00490	0.87262	0.02737
None	0.98166	0.00593	0.89268	0.02385

Figure 10: Test set accuracies over 5 generations of data set

Figure 10 shows the mean of test set accuracies and standard deviation over 5 generations of the data set for each depth.
Results for the first data set can make us conclude that a maximum depth of 8 seems the best. Indeed, the mean is the higher and the standard deviation the lower. However, to limit the complexity of the model and as the results are quite similar, a maximum depth of 4 would be better.
When looking at the mean for the second data set, an unconstrained depth is giving better results.

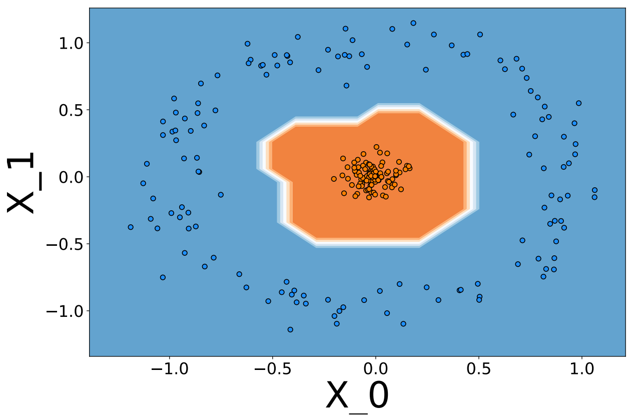
1.3 Difference between both problems

The main difference between both problems is that the second data set has some points that could be classify in both class. Indeed, there is some overlaps between the two ellipses introducing more difficulties to classify. A higher depth is needed in the second case because of that and the classification couldn't be as good as the first case. The first data set is aligned to the axis, this makes its classification easier. By the way, the mean of test set accuracies over several generations is way better for the first problem and also higher with a low depth.

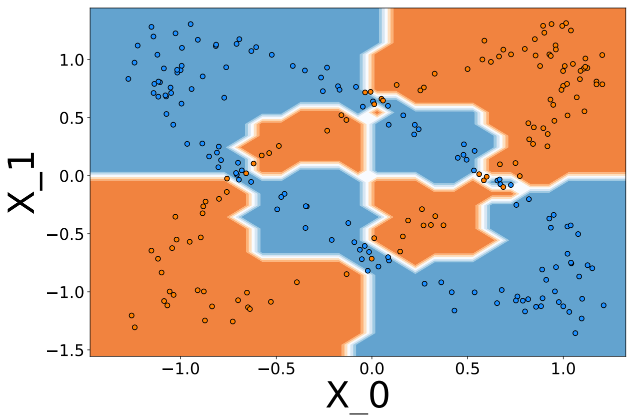
2 K-nearest neighbors

2.1 Decision boundary

a. Illustration

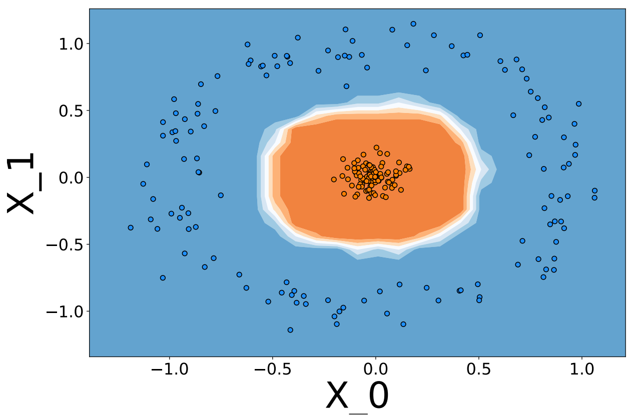


(a) Data set 1

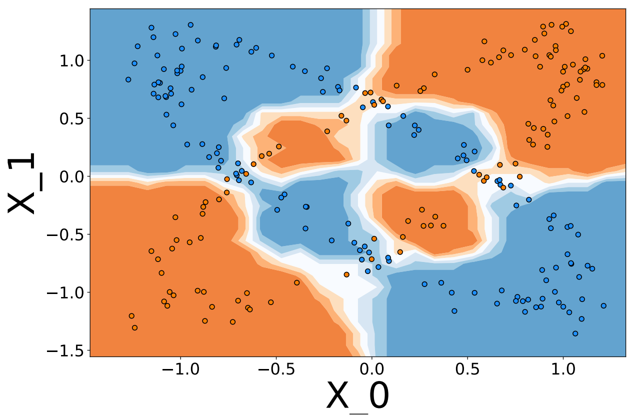


(b) Data set 2

Figure 11: Knn with 1 neighbor

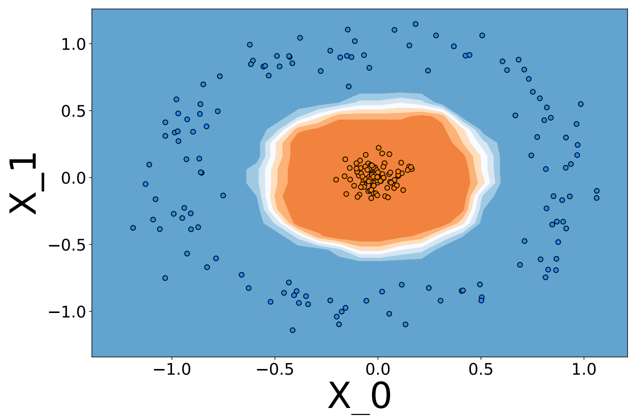


(a) Data set 1

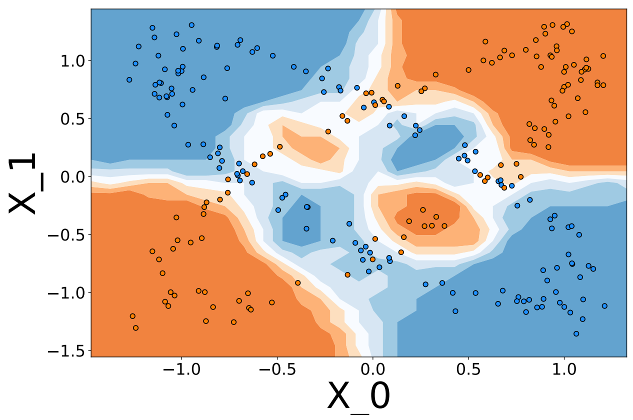


(b) Data set 2

Figure 12: Knn with 5 neighbors

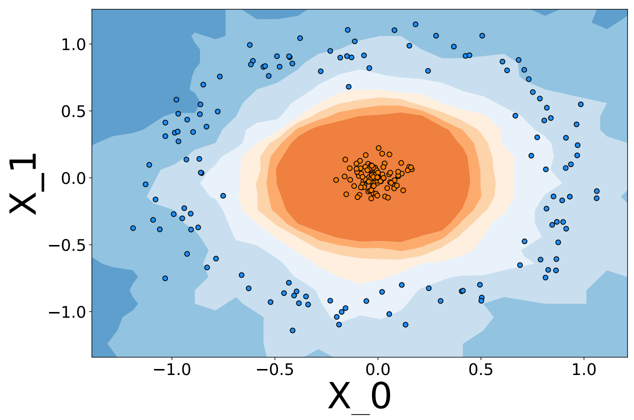


(a) Data set 1

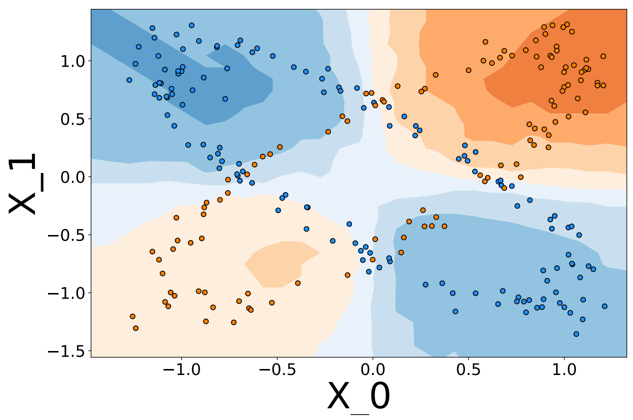


(b) Data set 2

Figure 13: Knn with 10 neighbors

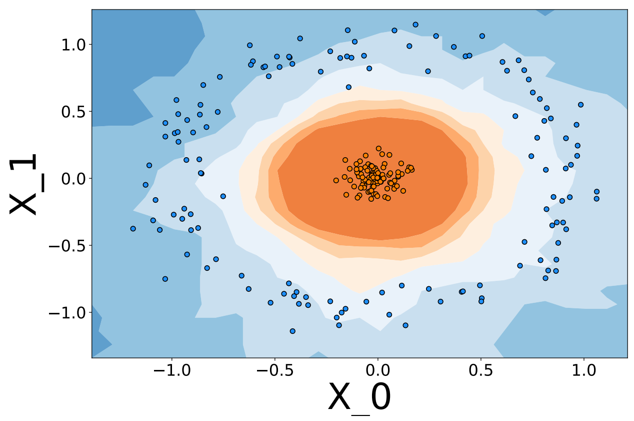


(a) Data set 1

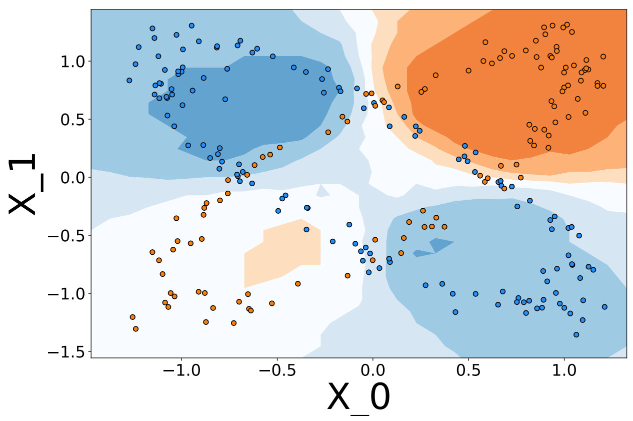


(b) Data set 2

Figure 14: Knn with 75 neighbors

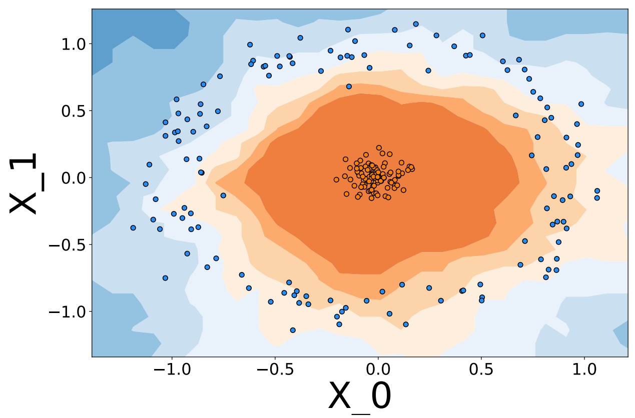


(a) Data set 1

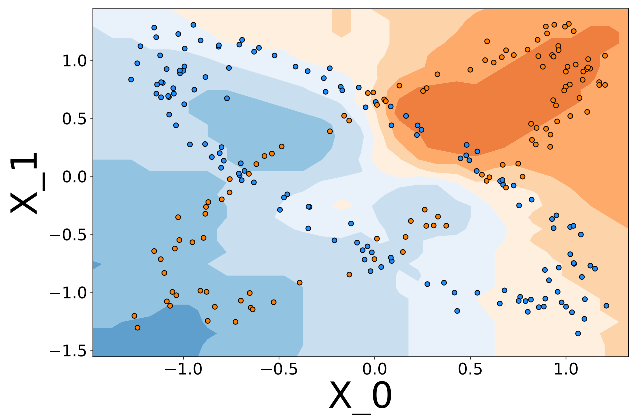


(b) Data set 2

Figure 15: Knn with 100 neighbors



(a) Data set 1



(b) Data set 2

Figure 16: Knn with 150 neighbors

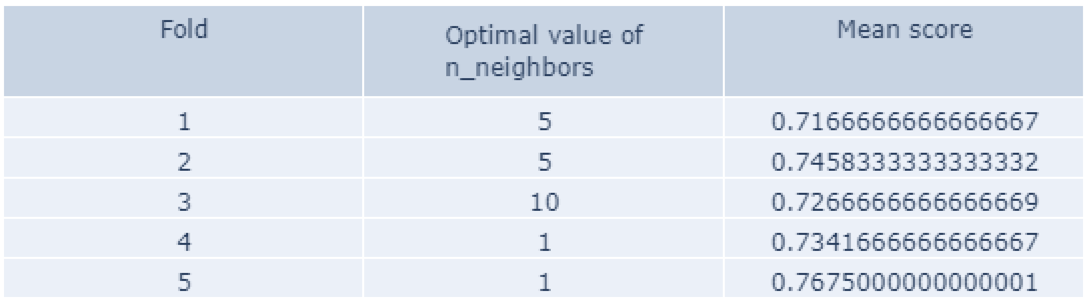
b. Evolution of the decision with number of neighbors

In figure 11 to 16 are illustrated the decision boundaries for both data sets in function of the number of neighbors in the model, i.e. for every observation we assign it the class of the k neighbors.
We clearly see that the accuracy is better with 1 neighbor, for both problems. Indeed, the model is confident in figure 11 because the boundary is sharp.
When the number of neighbors increases there are lighter colors meaning that the model is less and less confident about its predictions.
In the figures 12 and 13 the boundaries are still good but we got more zones with a lighter color that means that the model starts to have uncertainty.
In the next figures, that represents $k = \{75, 100, 150\}$ neighbors we do not have a confident model anymore. Also notice that $k = 100$ is already bad because some observations do not have classification and an area without observations is classified. The situation gets worst with $k = 150$ because for the data set 2 the classification is mostly erroneous.

2.2 Five-fold cross validation

a. Methodology

To implement the five-fold cross validation strategy, first, we split the data that we have, $x_{LS}, x_{TS}, y_{LS}, y_{TS}$ into 5 different subsets. For each fold i, $i \in \{1, .., 5\}$, we take as a training sample 4 of the subsets and set the missing subset as the cross validation data. We calculate the prediction values for the response variable and obtain the accuracy score between the predicted value and the real value of the variable. After, the mean of the scores is calculated for each fold. This process is applied to the different number of neighbors in $\{1, 5, 10, 75, 100, 150\}$



Fold	Optimal value of n_neighbors	Mean score
1	5	0.7166666666666667
2	5	0.7458333333333332
3	10	0.7266666666666669
4	1	0.7341666666666667
5	1	0.7675000000000001

b. Optimal value of neighbors

Given the results obtained in the illustration of the decision boundary graphs, we have concluded that the optimal value of the K-neighbors is between 1 and 10, it can be noticed that the most added neighbors the less accuracy we have in the boundaries (we can see this because of the color in the graphs, the lighter the color is the lower the confidence of the model).
This result was corroborated by the mean score over the five folds, for each fold we obtained that the model with the lowest error was the one corresponding to 1,5 or 10 neighbors and the more neighbors we have the least accuracy we get of the model. When the final data test set is independent of all the learning sets used in cross validation we obtained an unbiased estimate for the test accuracy.

2.3 Optimal value of neighbors in function of the LS size

a. Evolution curve of test set accuracies

Following, the plots showing the evolution of test accuracies on a training sample of size 500, these for every possible possible number of neighbors for learning samples of sizes $\{50, 200, 250, 500\}$. We have observed that the number of neighbors can at most be equal to the sample size and no greater than 200.

Data Set 1

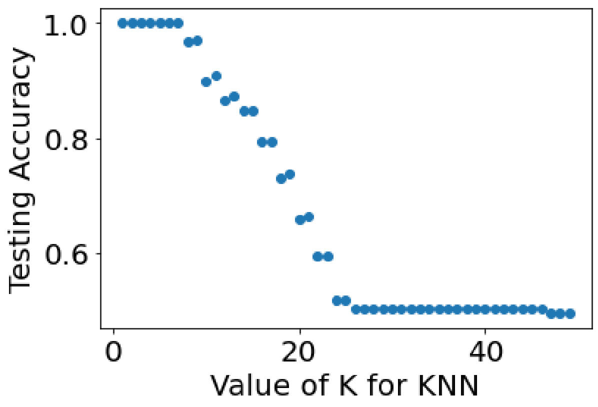


Figure 17: Learning sample of 50

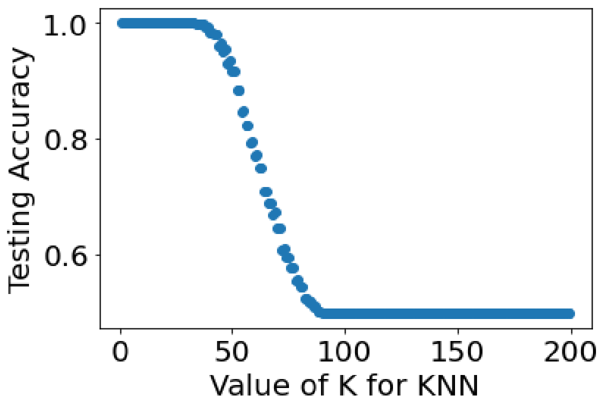


Figure 18: Learning sample of 200

We can observe that, first, the largest the sample is the bigger the accuracy is. In the figures 17 and 18 we observe that values between 1-10 and 1-75 neighbors give more accuracy, while

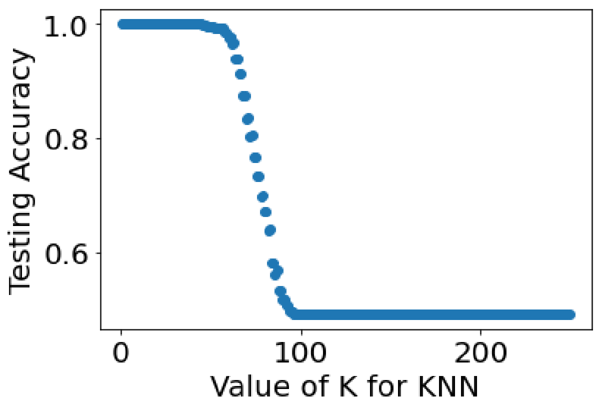


Figure 19: Learning sample of 250

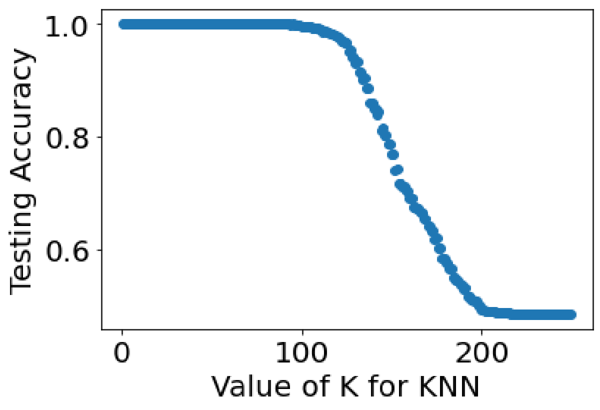


Figure 20: Learning sample of 500

a bigger number of neighbors reduces too fast the accuracy and returns one that is too close to zero. In the next two figures we observe a similar behavior, the accuracy is greater if the number of neighbors is small.

Data Set 2

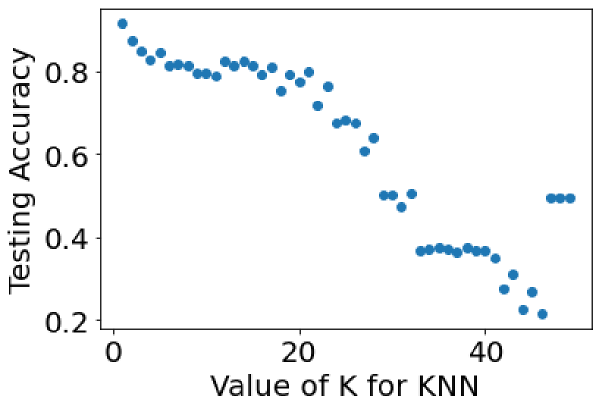


Figure 21: Learning sample of 50

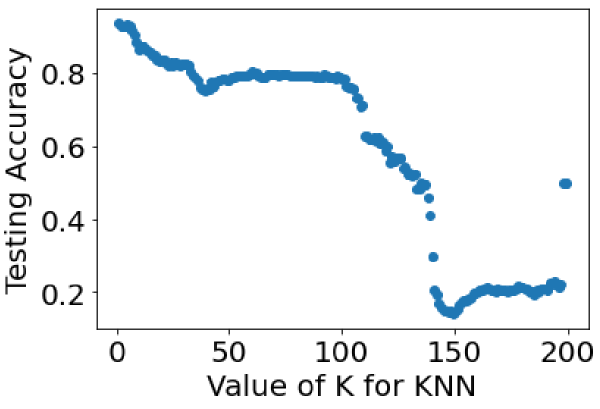


Figure 22: Learning sample of 200

Figure 23

We observe a similar behavior as the one from the data set 1. The least number of neighbors the greater the accuracy of the model. The main difference is that a biggest number of neighbors with a accuracy greater that 0.5 are accepted. For 100 neighbors we have a good accuracy in the model, grater than 0.7, we can observe than if the quantity of neighbors is greater that 100 the accuracy reduces its value and it becomes way smaller, ie, the decreased due to under-fitting.

b. Optimal value of neighbors respect to the LS size illustration

We have almost the same results for both data sets and for all the different sizes of the Learning sample. Thus, with the two data sets provided we have that the optimal number of neighbors corresponds to 1, or 5 in the case of the data set 1. This means that an observation gets classified in the class of its nearest neighbor, this closeness is measured with the euclidean distance, so

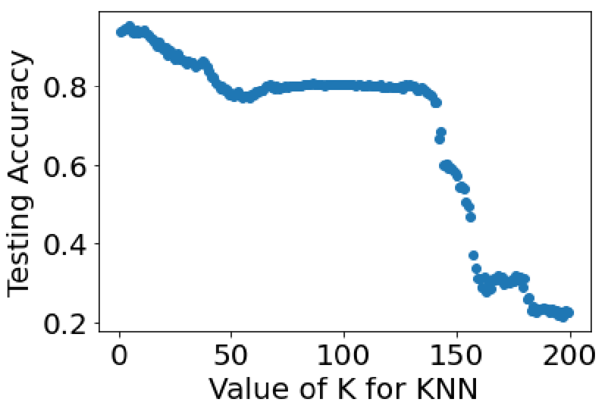


Figure 24: Learning sample of 250

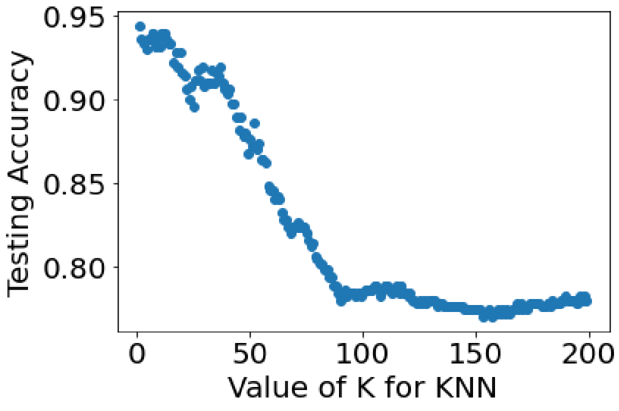


Figure 25: Learning sample of 500

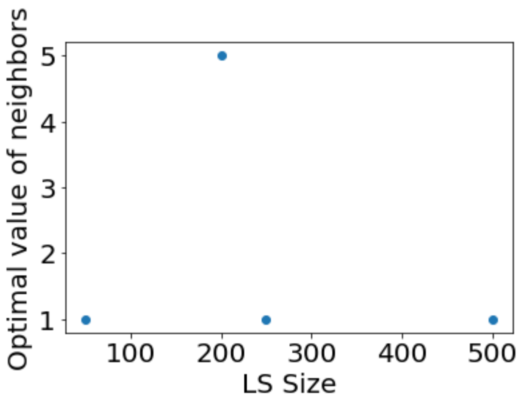


Figure 26: Data Set 2

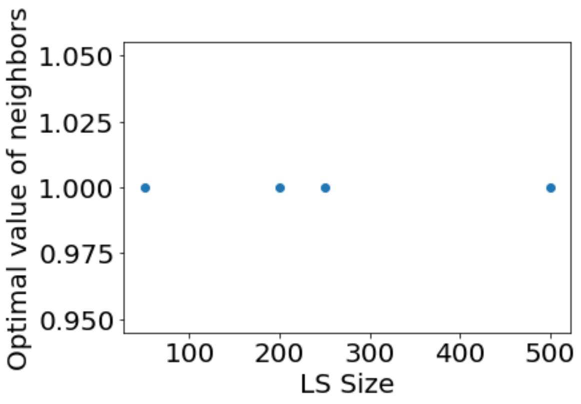


Figure 27: Data Set 1

the fact that the optimal number of neighbors is one in the data set 1 is reasonable because all the data is grouped in the same space, two circular zones independent one of another. For the data set 2 we also have both classes separated, nevertheless there is a zone in which the data points of each class gets closer to each other so this can lead us to a bigger number of neighbors, in this the case using a LS of size 200.

2.4 Five-fold cross validation discussion

The results obtained in question 2.3 for the values of a learning sample of size 250 and the results obtained by using five fold cross validation were the same, given this fact considering that there are no pre-defined statistical methods to find the most favorable value of neighbors and that the time complexity of using the cross validation algorithm is higher, we would suggest the standard calculation of the accuracy. Nevertheless, the k-folds cross validation has the advantage that it increases the generalization of the model because it let us build K different models, this allows to make predictions on all the data. Another advantage is that cross validation prevents the model from overfitting the training data set due to the way it divides the data set.

3 Residual fitting

3.1 Algorithm demonstration

In the Residual Fitting (Forward-Stagewise Regression) algorithm that is a simple algorithm to fit iteratively a linear regression model. Proof that the best weight w_k for the attribute a_k introduced in the model at step k is $\rho_{ak}, \Delta_{ky}\sigma_{\Delta_{ky}}$, where $\rho_{ak}, \Delta_{\Delta_{ky}}$ is the Pearson correlation between a_k and Δ_{ky} and $\sigma_{\Delta_{ky}}$ is the standard deviation of Δ_{ky}.

Proof. We have that the coefficients in a linear regression are of the form

$$w_1^* = \frac{\sum_{i=1}^{N}(a_1(o_i)-\bar{a}_1)(y(o_i)-\bar{y})}{\sum_{i=1}^{N}(a_1(o_i)-\bar{a}_1)^2},\ w_0^* = \bar{y} - w_1^*\bar{a}_1$$

In this case, we have a regression of Δ_{ky} over a_k and the characteristics have zero mean and variance of one. This implies that $\sum_{i=1}^{N}\left(a_1\left(o_i\right)-\bar{a}_1\right)^2 = 1$ and we have that

$$\sum_{i=1}^{N}\left(a_1\left(o_i\right)-\bar{a}_1\right)\left(y\left(o_i\right)-\bar{y}\right) = cov(a_k, \Delta_{ky})$$

thus,

$$\sum_{i=1}^{N}\left(a_1\left(o_i\right)-\bar{a}_1\right)\left(y\left(o_i\right)-\bar{y}\right) = \rho_{ak}, \Delta_{ky}\sigma_{\Delta_{ky}}$$

And $\rho_{ak}, \Delta_{ky}\sigma_{\Delta_{ky}}$ is the optimal weight for w_k because is the optimal coefficient according to the Minimun Gaussian Squares.

□

3.2 Algorithm implementation

In the function fit of residual_fitting.py, the algorithm has been implemented. For the original case where X has 2 attributes, we compute a weight matrix of 3 elements. But the algorithm has been generalized so that we can use it with a higher number of attributes.

3.3 Residual fitting model

a. Decision boundaries illustration

In figure 28 and 29 is represented the decision boundaries with a residual fitting model.
As we can see, the classification is quite bad. This could be explained by the fact that a linear regression with two attributes divides the data in regions determined by a right line. By the way, this model is not adapted to classify data in circles or ellipses.

b. Test accuracies

The test accuracies are respectively 30.4 and 54 % for data set 1 and 2. As previously said, the classification seems quite bad with this model.

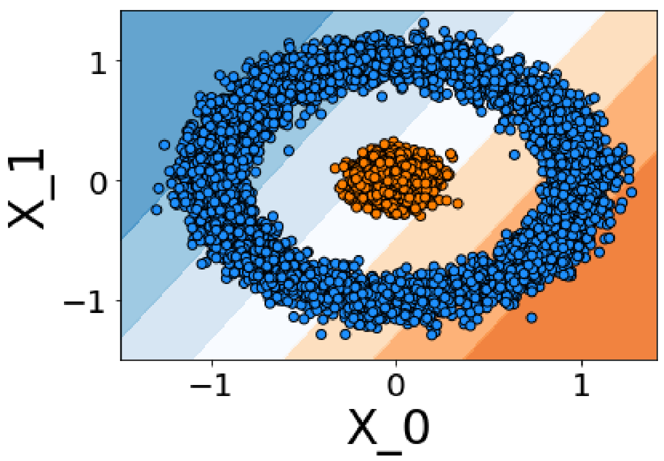


Figure 28: Decision boundaries for data set 1

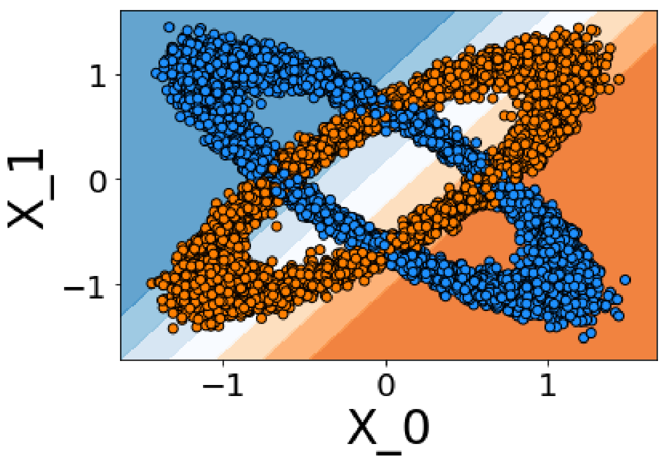


Figure 29: Decision boundaries for data set 2

3.4 Residual fitting model with new attributes

a. Decision boundaries illustration

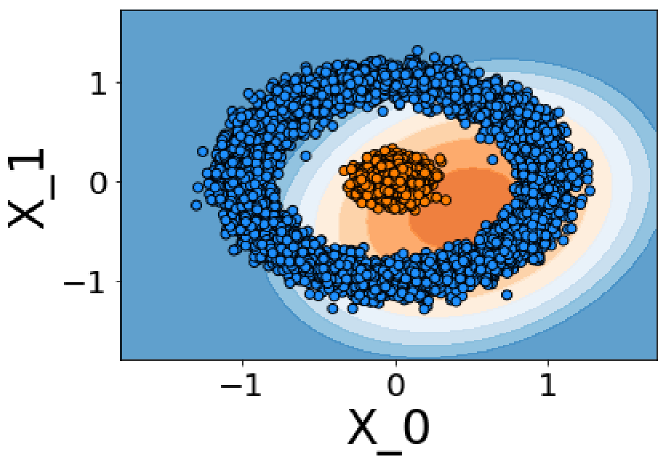


Figure 30: Decision boundaries for data set 1

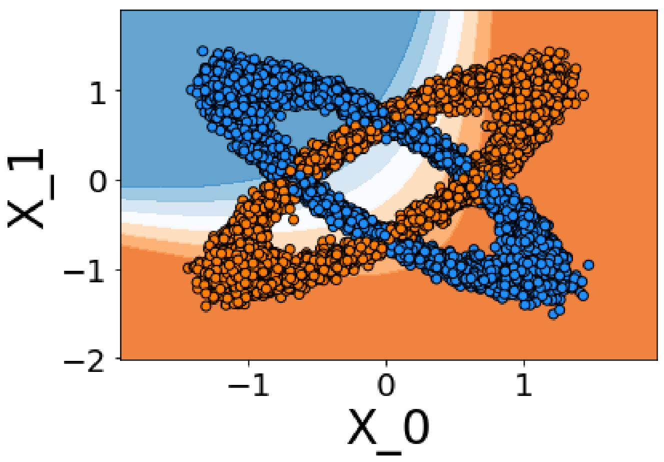


Figure 31: Decision boundaries for data set 2

In figure 30 and 31 is represented the decision boundaries for the model fitted with an extended attributes matrix. In this case, 5 attributes are used instead of 2. Differences with the previous model can be seen in the illustrations, we notice that the classification is no longer made according to a straight line. Indeed, the classification is still made in two regions but these regions are separated by a curve.

b. Test accuracies

The test accuracies are respectively 30.12 and 48.96 % for data set 1 and 2. These results are similar to the ones found with the previous linear model. However, in figure 30, the classification would have been way better if the decision boundary would have shift a little. Maybe some errors have introduce a shift and so decreased the accuracy of the model. In conclusion, this extented model doesn't give better results but is seems that some improvements could make it more efficient than the original one.