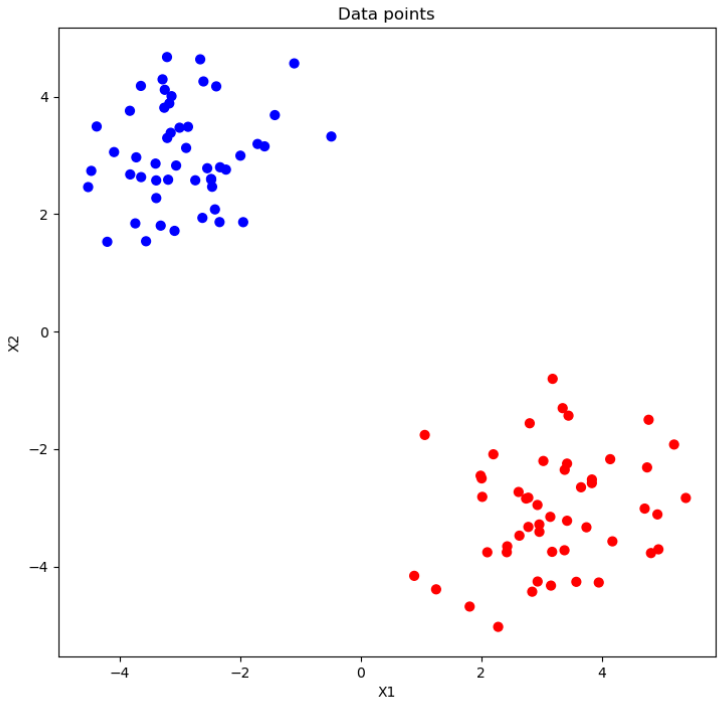
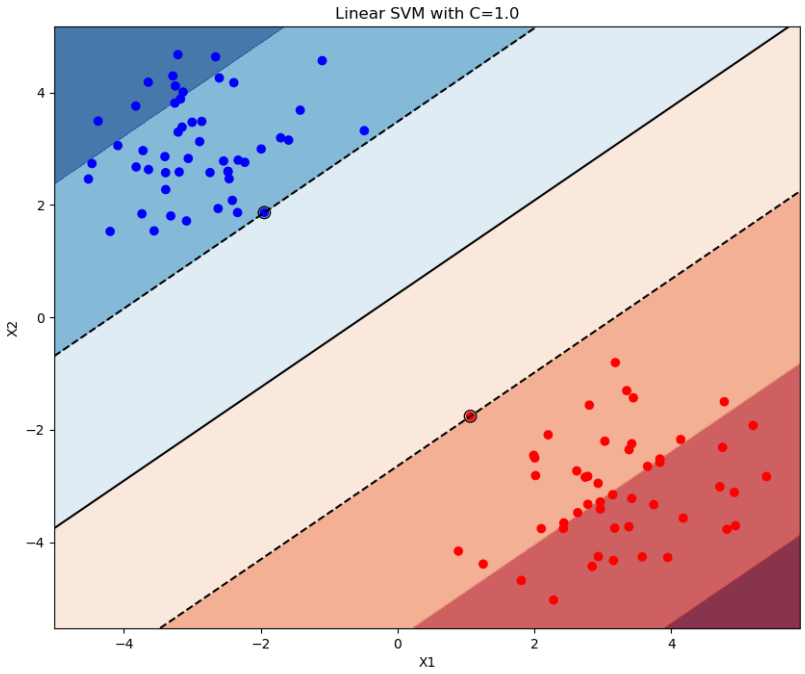
Linear SVM

Here below the graph representing the dataset used for this section where different colors correspond to different labels.



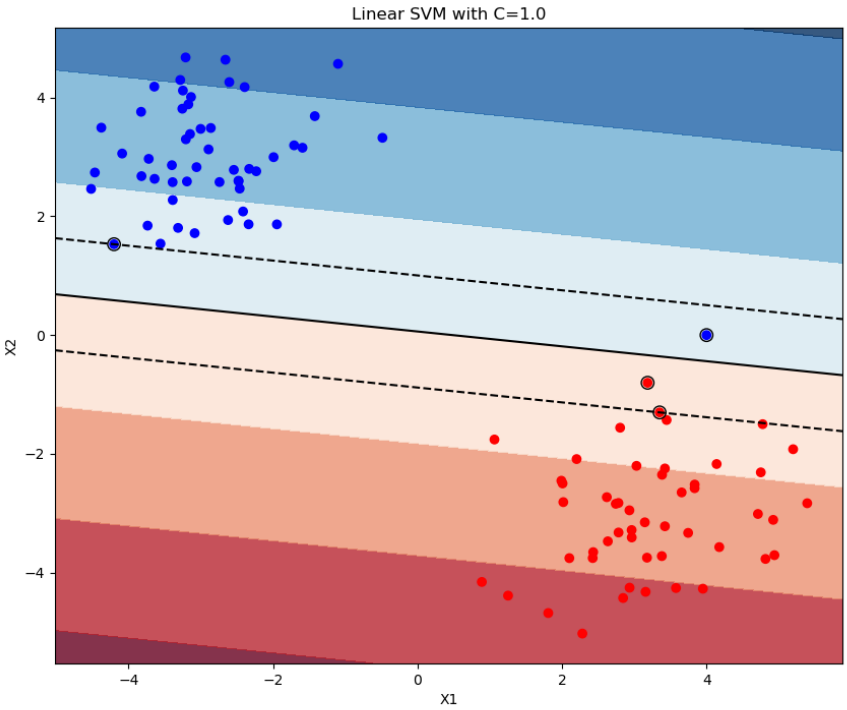
Ex_1_a

Here below the graph showing the result of the SVM trained with a linear kernel and the provided dataset.



1.2 Ex_1_b

The point added in this section is the blue point that is very far away from the others at the cords (4, 0) (label 1 so blue). As we can see this point is part of the blue group and not the red one it seems closer to. As we can see this point, being counted in the "blue dataset", leads to a modification of the decision boundary since it seems clear looking at the graph in section 1.1 that with the previous decision boundary would instead fall within the red group of points.



1.3 Ex_1_c

As we can see in the graphs below, a high value of C contributes to a higher accuracy in the choice of decision boundary. In the chart resulting from C = 10^6 we can see that all points are on the right side of the decision boundary and there is no point between the margin (dashed line) and the boundary decision so we can consider all the points as support vector.

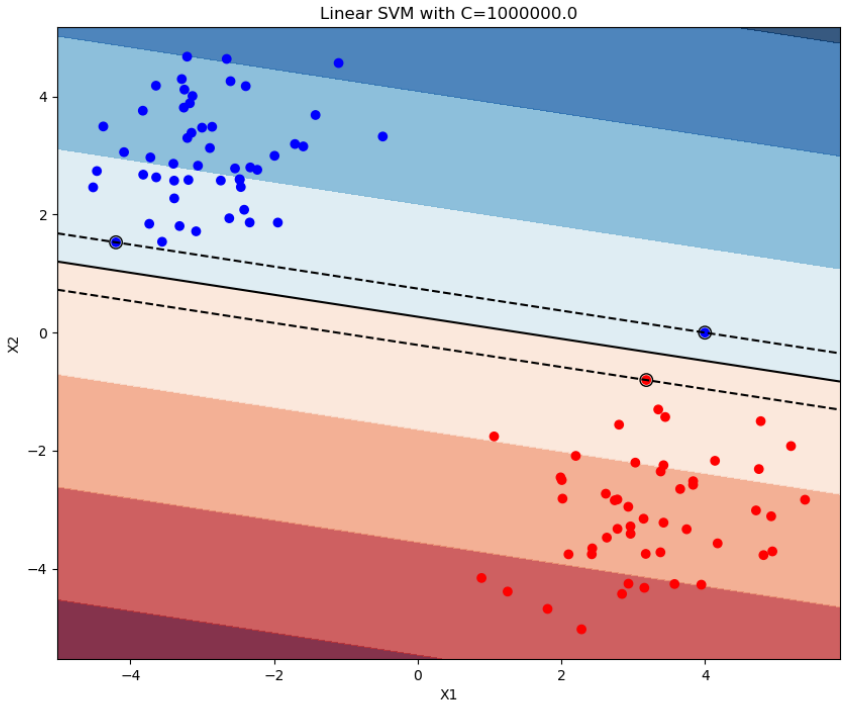
The second graph is the same graph of section 1.2 and present no points on the wrong side of the decision boundary but 2 points (one red and one blue) on the maximum-margin hyperplane (the two parallel hyperplanes that separate the two classes of data).

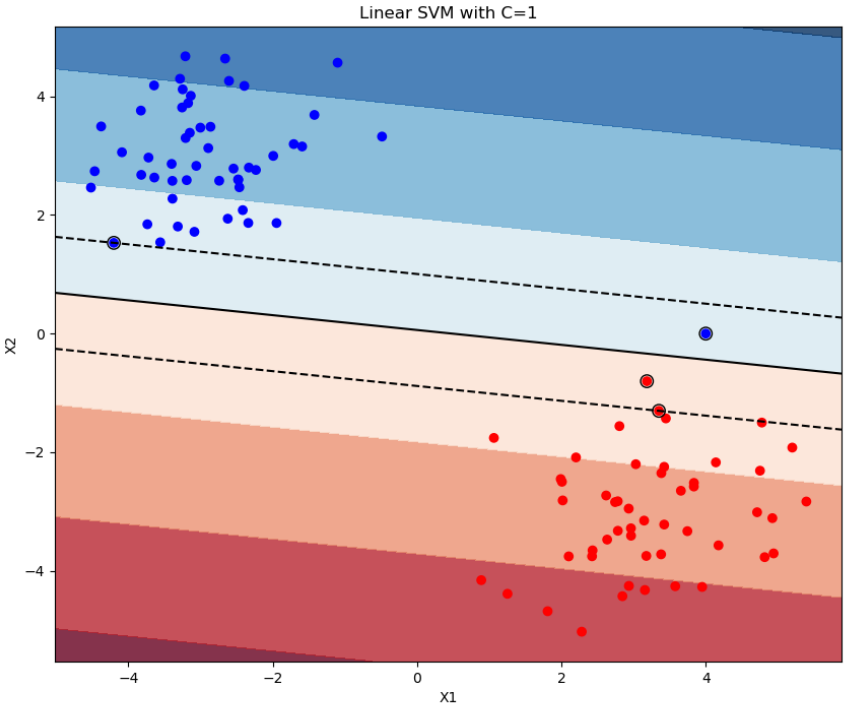
The third graph where C = 0.1 has instead a blue point (the added one) on the wrong side of the boundary (the red one).

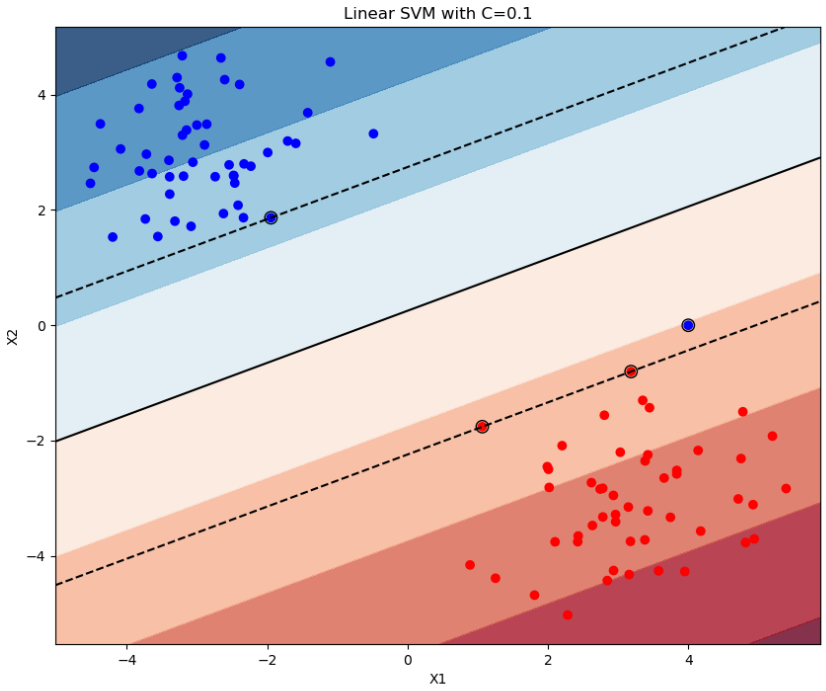
Also the last graph has the added point on the red side of the boundary and presents a lot of points of both sides that are on the maximum-margin hyperplane.

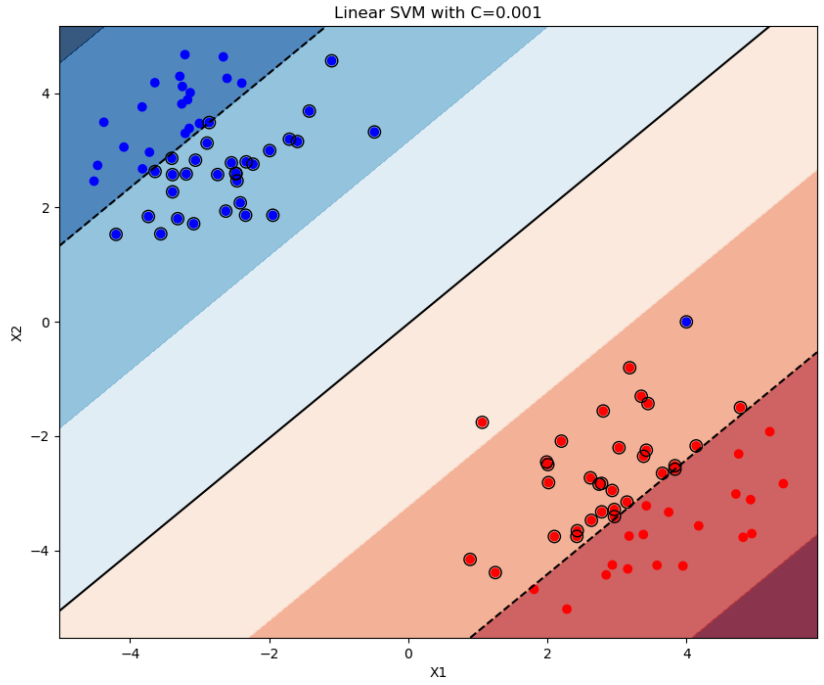
Interesting is to see that for C > 1 and C < 1 the slope of the boundary changed direction.

In this case we have seen that with a high value of C all the points result in support vector while with a too low value many points end up inside the maximum-margin hyperplane.


Linear SVM with C=1000000.0
X1
X2
4
2
0
−2
−4
−4
−2
0
2
4

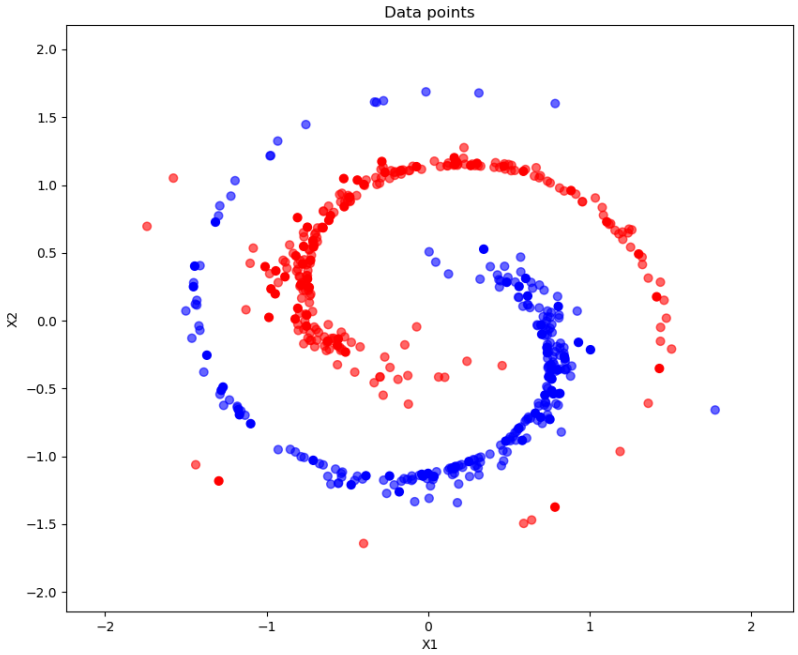

Linear SVM with C=1
X1
X2
4
2
0
−2
−4
−4
−2
0
2
4


Linear SVM with C=0.1
X1
X2

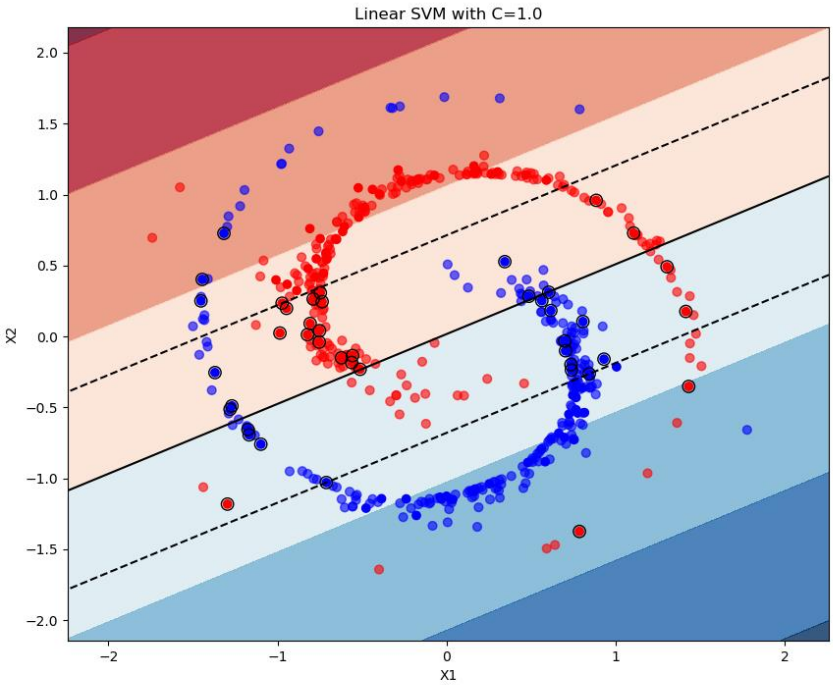

Linear SVM with C=0.001
X1
X2

1 Nonlinear (kernel) SVM

Here below the graph representing the dataset used for this section.

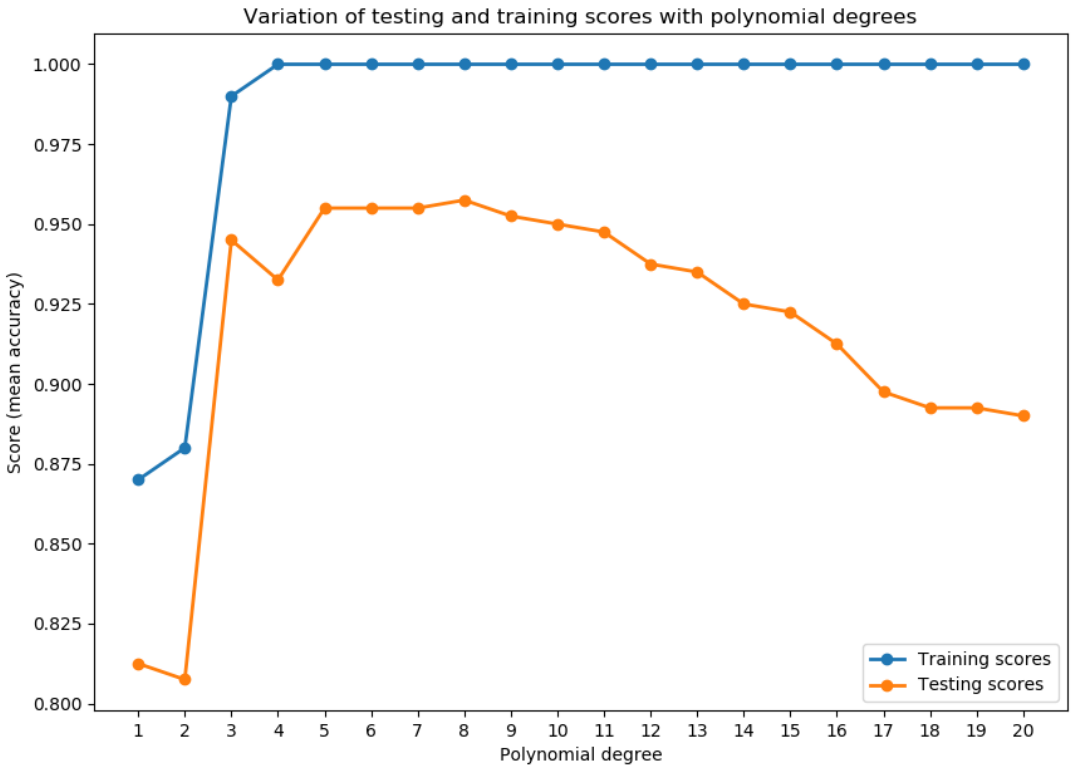


1.1 Ex_2_a – Linear kernel

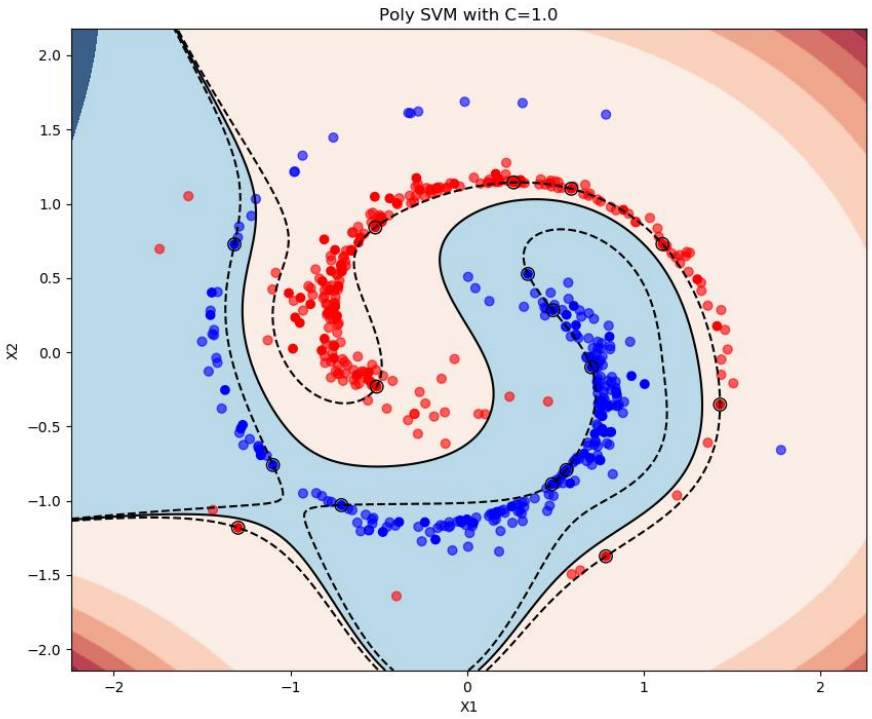


As we can see from the graph above the linear kernel is probably not the optimal kernel for this type of dataset which is very difficult to split into two sections with a linear boundary. The mean accuracy of classification (obtained with the method score) for the testing set is 0.815 which is high but for sure not optimal.

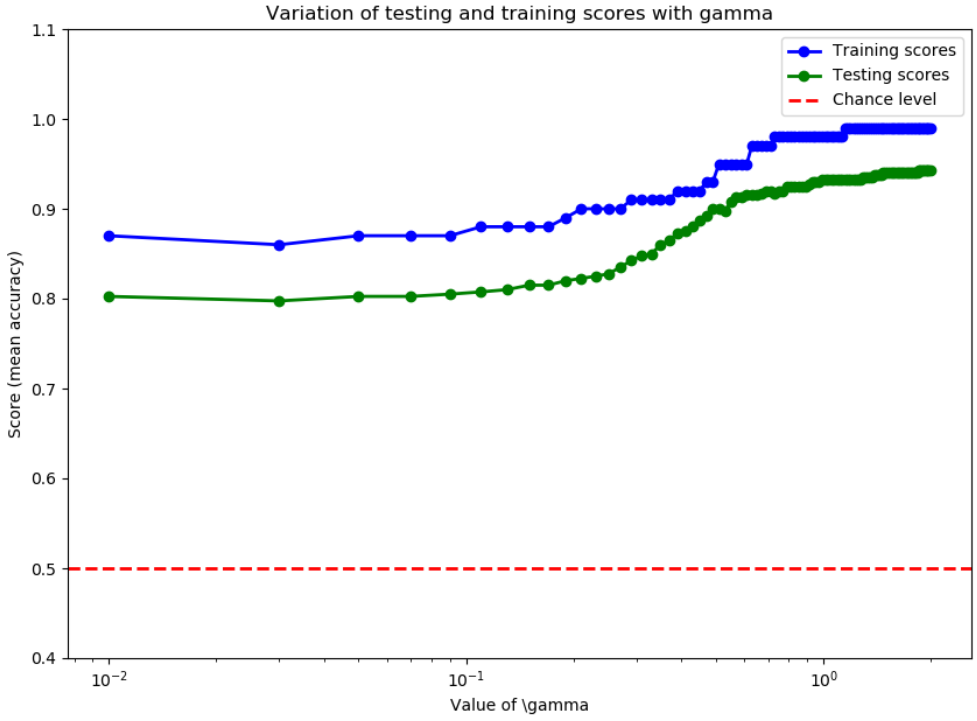
1.2 Ex_2_b - Polynomial kernel



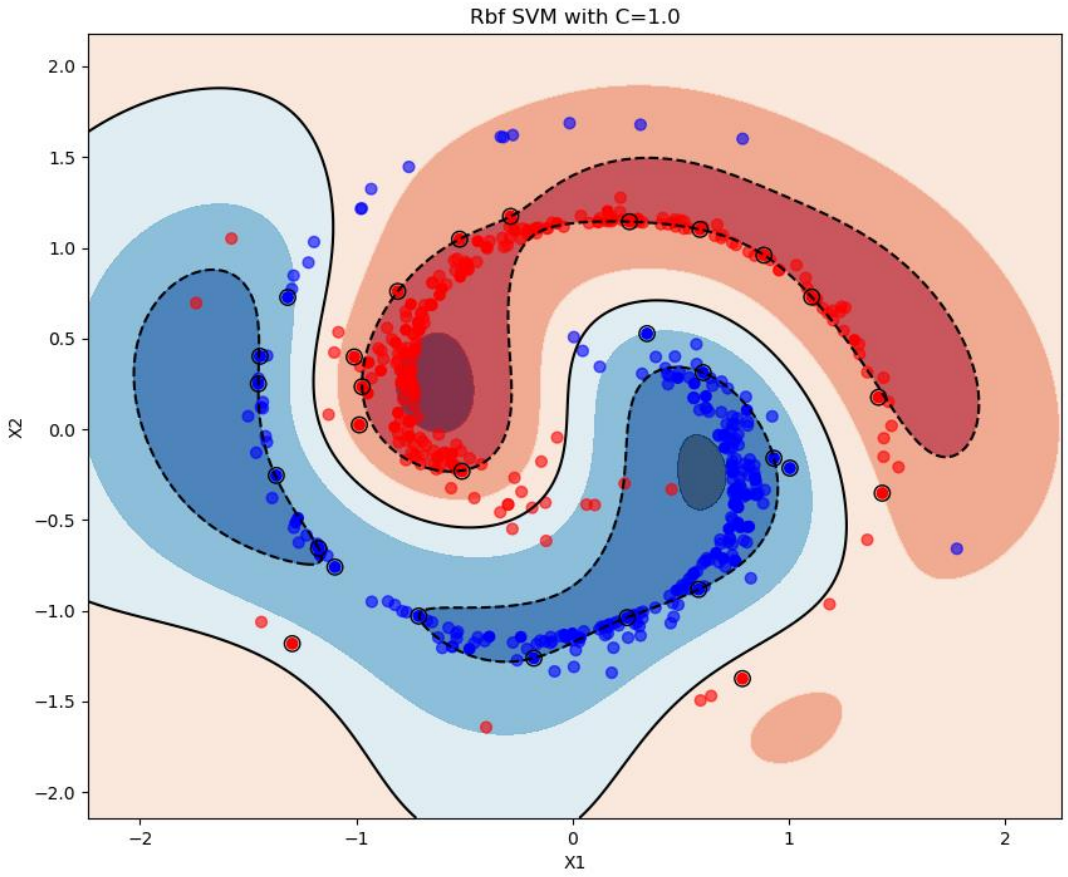
The graph above shows the mean accuracy of classification for each degree for both training and testing set. The optimal polynomial to describe this data set is the grade 8 polynomial shown below that presents an accuracy of 0.955 that is a lot better than the one obtained with the linear kernel. Interesting to see is also that the training set reach an accuracy of 100% already with a polynomial of grade 4 and then it stays costant.



Ex_2_c – RBF kernel



The graph above shows the mean accuracy of classification for each value of gamma for both training and testing set. The optimal result is obtained with a gamma value of 1.8499e and has an accuracy of 0.9425. The graph for the optimal gamma is shown below.



1.3 Observations

- The polynomial kernel is performing the best as it has the best optimal score.
- As we can see from the graphs the more the decision boundary is complex more are the founded support vectors
- The RBF kernel is generalizing better than the others kernel as the behavior of the boundary seems more similar to the behavior of the points than the one of the other kernels

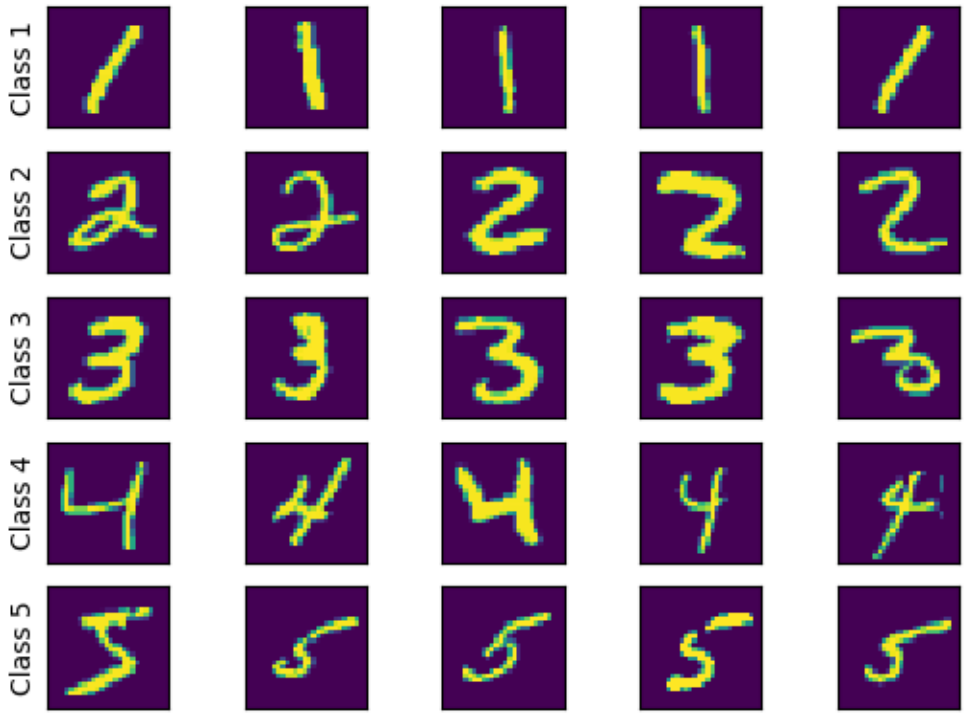
2 Multiclass classification

One versus rest

The multi-class dataset is splitted into multiple binary classification problems. A binary classifier is trained on each binary classification problem and the with the most confident model the predictions are made. [1] One binary classifier needs to be trained for each class.

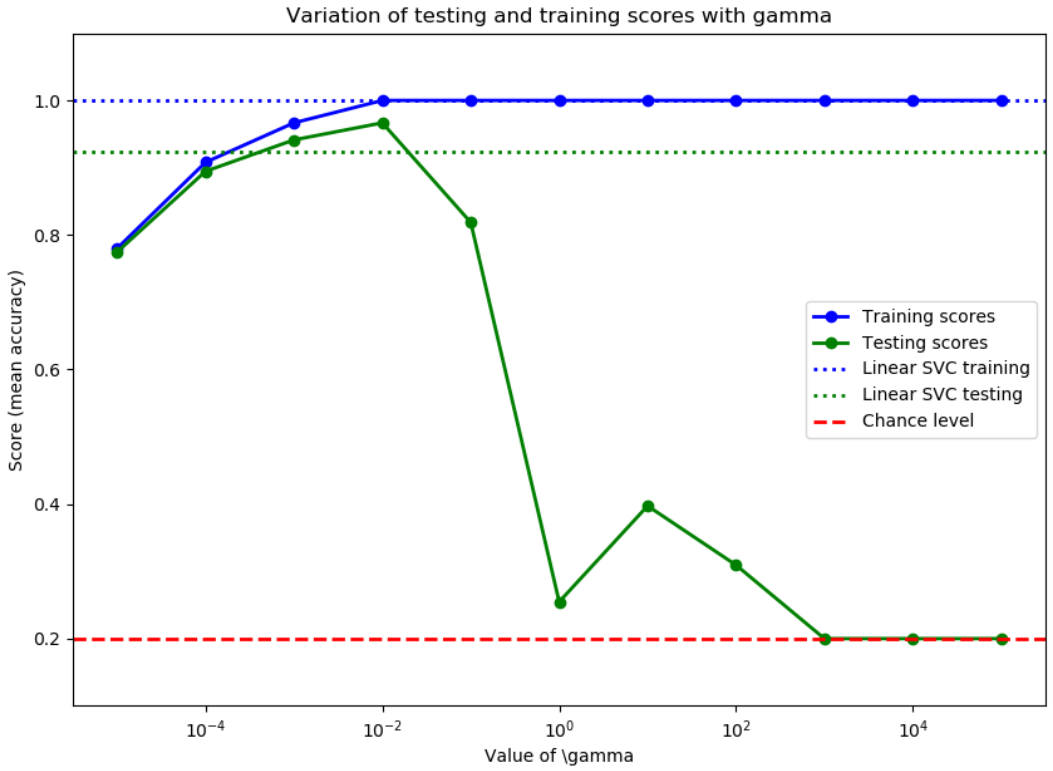
One versus one

The multi-class dataset is splitted into one binary dataset for each class versus every other class. The model with the most predictions or votes is predicted by the one-vs-one strategy.[1] For each class n-1 classifiers must be trained.



Some examples of the data that is used for this example

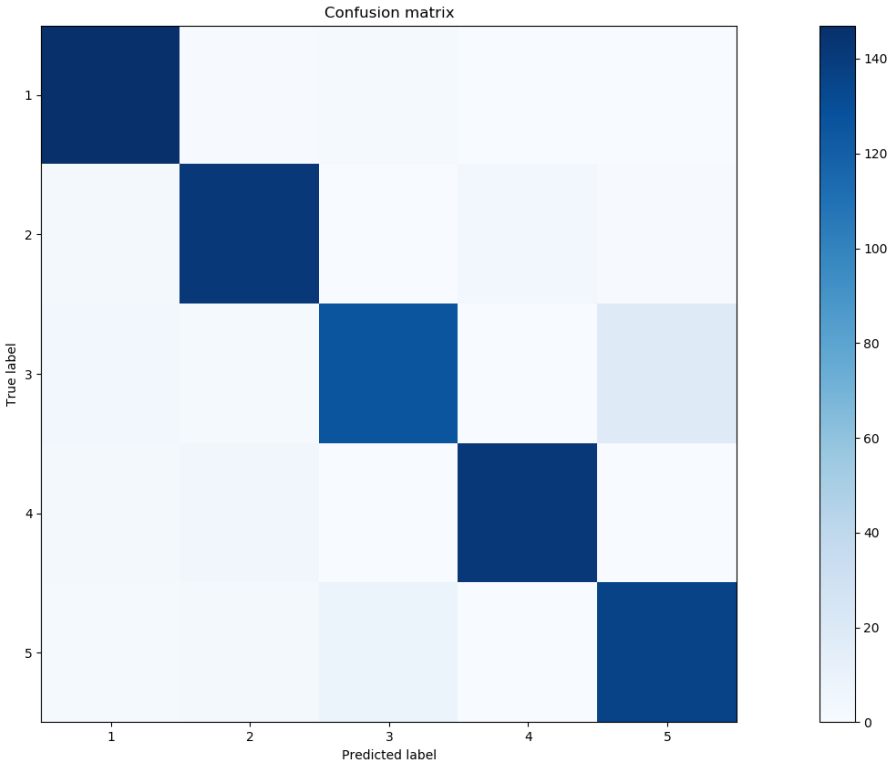
[1] https://machinelearningmastery.com/one-vs-rest-and-one-vs-one-for-multi-class-classification/



Scores for a linear and a rbf kernel

In the above figure, which shows the training and testing scores for a linear and a rbf kernel we can see that even though the training score of the rbf kernel increases with a higher gamma up to 1, the testing scores decreases with rising gamma. Meanwhile the linear kernel produces a relatively good testing score. Only for gammas 10^{-3} and 10^{-2} is the rbf testing score higher than the linear testing score.

3.b



Confusion matrix

The most digit class with the highest error rate is class 3. There we get the values: [4 2 126 0 18]

Here we see that of all the digit that are predicted to be a “3”, 126 are predicted correctly, while 4 are actually a “1” , 2 are a “2” and 18 actually are a “5”.

The first 10 misclassified digits are shown in the figure below.

When looking at the 5 we can see that the bottom part of the 5 is very similar to the bottom part of a 3. This is the reason why it gets misclassified. Meanwhile the 1 bears some resemblance to the backside of a 3 and because of that resemblance it sometimes gets misclassified.

4. SVM with Gradient Descent

In this example we have chosen a learning rate of 0.01 and a max_iter of 100.

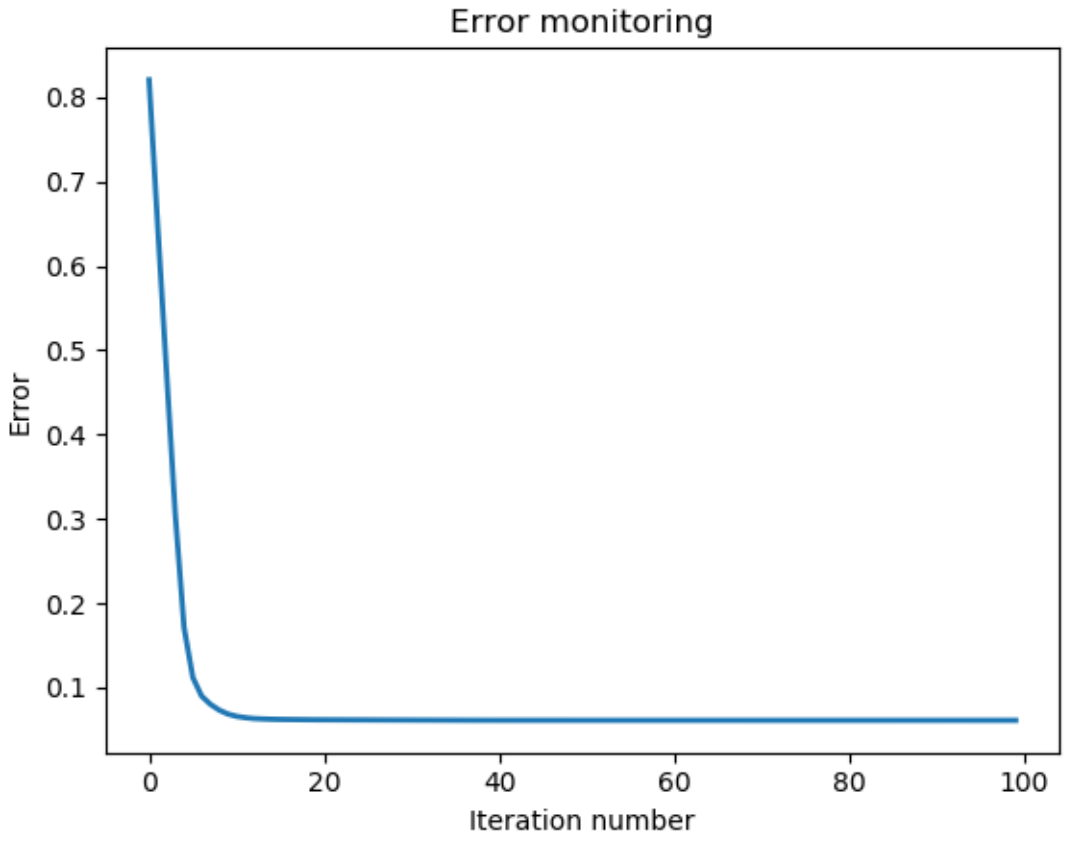
We were able to obtain the following parameters

cost [0.06119312]

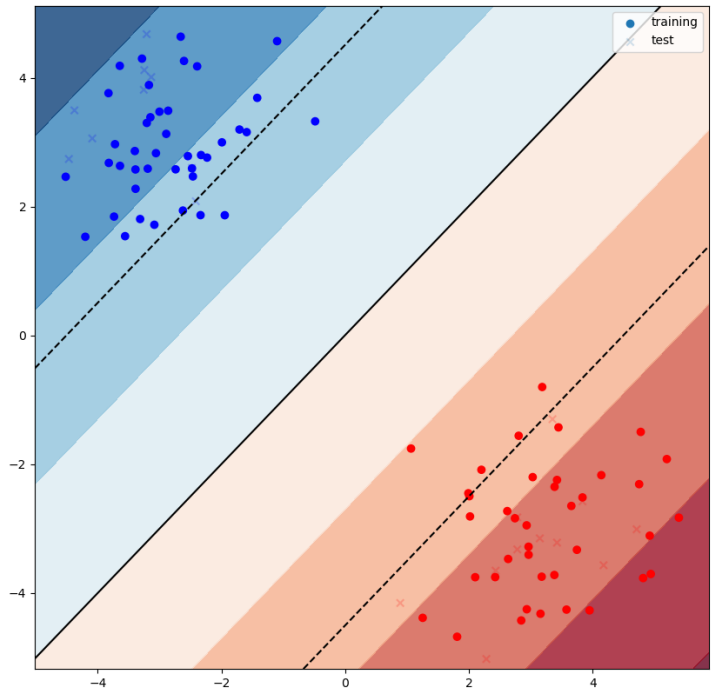
w optimal [-0.22215373 0.22181264]

b optimal [-0.001]

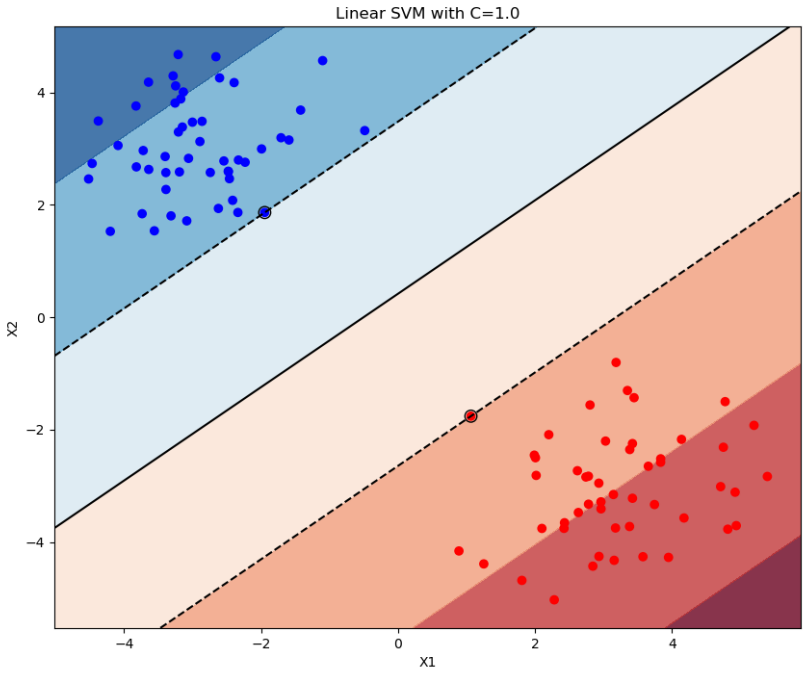
whilst having an accuracy of 1 on the test set.



Error over iterations



decision boundary of the gradient descent



decision boundary of the linear SVM

The first plot of the two plots above is the one using the gradient descent and the second one uses the linear SVM. We see that the decision boundary of the gradient descent is a bit steeper than the one with the linear SVM. Other than that they look rather similar.

The main drawback of minimizing the SVM is that the margin gets maximized. So there a trade-off has to be made between having less misclassifications and a higher margin, or allowing more some misclassifications to happen and keeping the margin small.