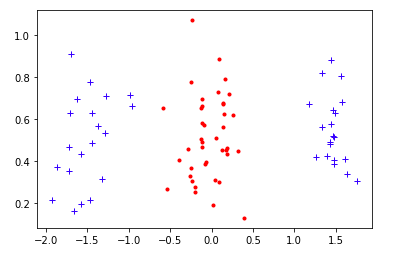
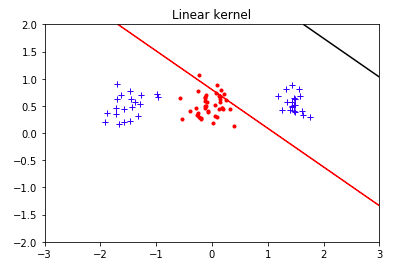
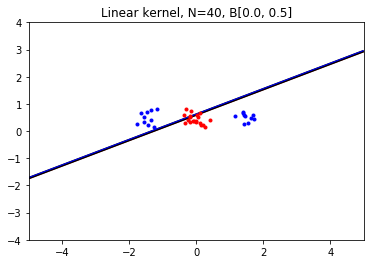


1. Move the clusters around and change their sizes to make it easier or harder for the classifier to find a decent boundary. Pay attention to when the optimizer (minimize function) is not able to find a solution at all.

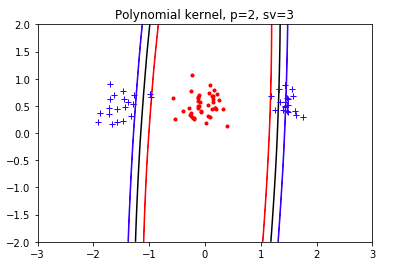
We created a new overlapping dataset

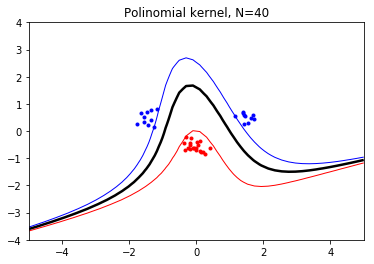
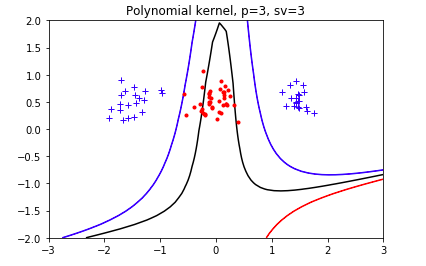


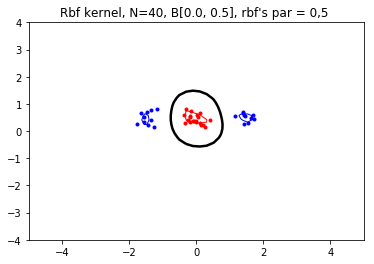
The linear kernel cannot find a hyperplane

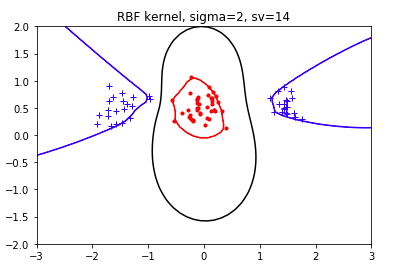


Try polynomial kernel





Radial kernel

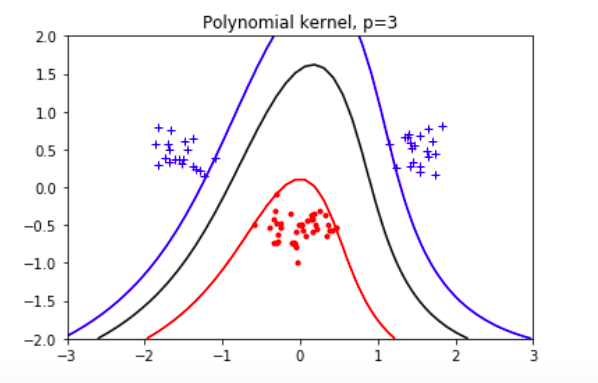
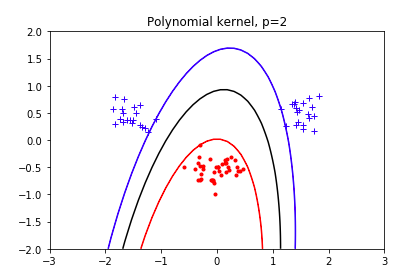


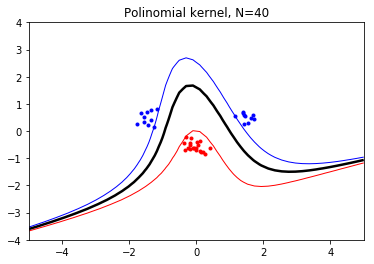
1. Implement the two non-linear kernels. You should be able to classify very hard data sets with these.
2. The non-linear kernels have parameters; explore how they influence the decision boundary. Reason about this in terms of the bias- variance trade-off.

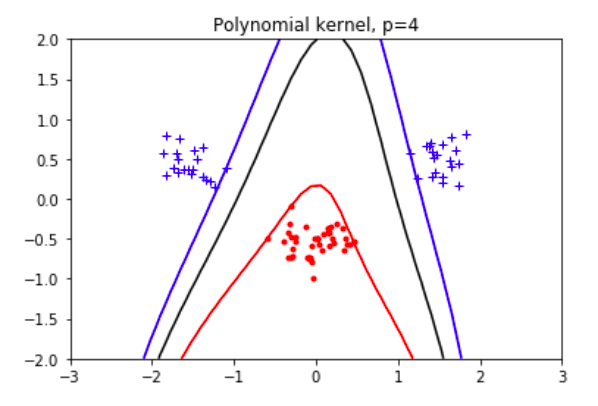
Polynomial kernel:

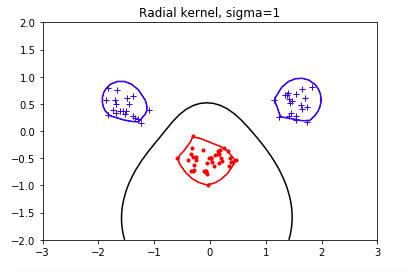
=0 is fail to find a boundary

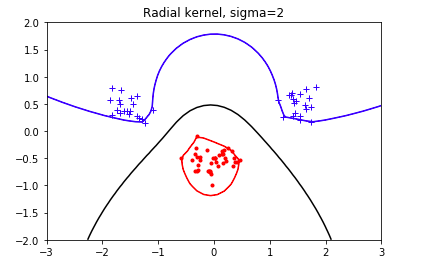
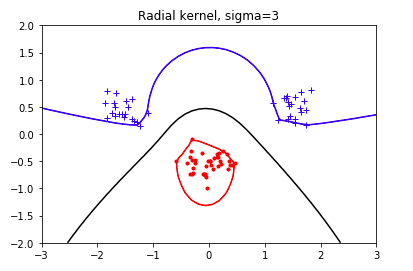
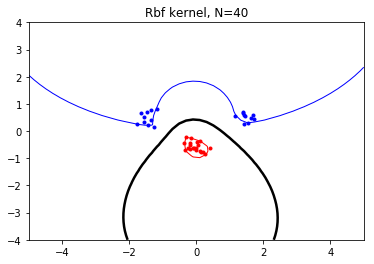
=1 is the same as linear kernel

The higher the p, the shape of the plane is more complex





RBF kernel: the higher the sigma, the less complex the boundary, fewer support vectors



Sigma= 0 every points are support vector

1: 13 sv

2: 10 sv

3: 10 sv

4: 10

6:10

The more complex model, the less likely to be biased but higher variance

1. Explore the role of the slack parameter C. What happens for very large/small values?

The higher the C, the smaller the margin, the fewer support vectors.

The very large C is closed to no upper bound for alpha

1. Imagine that you are given data that is not easily separable. When should you opt for more slack rather than going for a more complex model (kernel) and vice versa?

If the data is noisy, the lower C we should use to allow a larger margin, and the model is less complex.