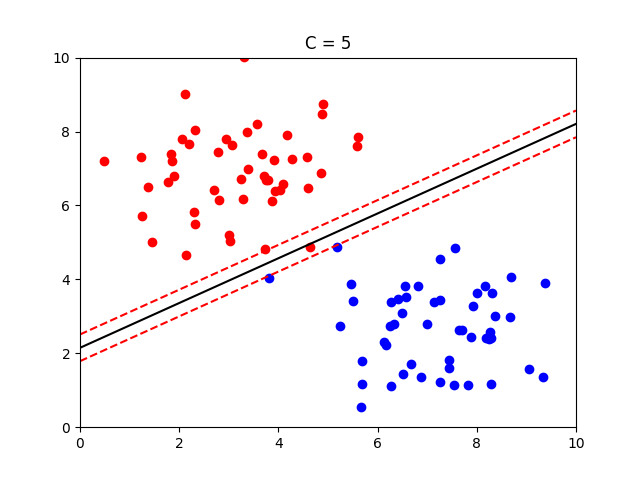
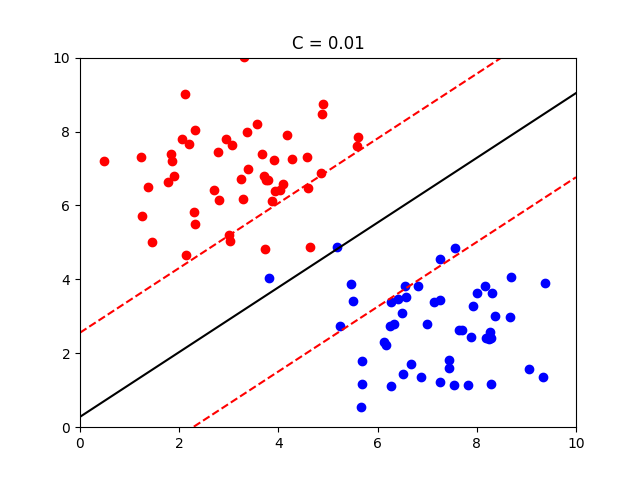
**Week 11 Worksheet: Support Vector Machines**

In these questions, you will be presented with data which we are trying to analyse and separate into red and blue, and in doing so creating a model which will allow the reliable classification of unseen data.

**Part A – Hardness/softness of margins**

These two graphs show the same set of data which has been fit with a linear kernel SVM, using different parameters to change the hardness of the margin.

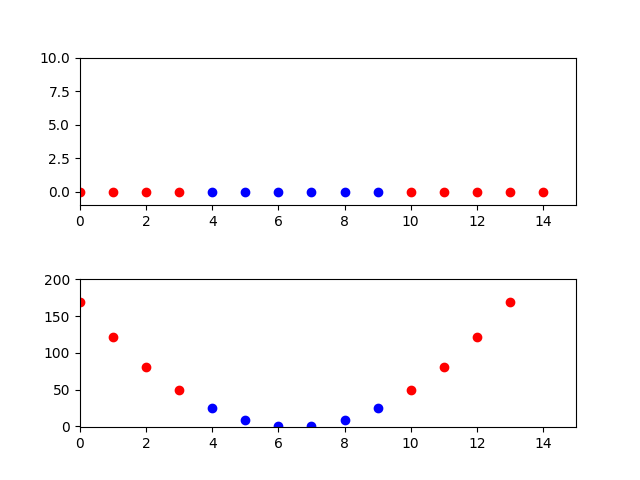
Model A Model B



Explain the advantages of using Model B, with reference to:

* hard and soft margins
* how well the model will respond to new data
* overfitting/underfitting

Model B is advantageous for future classification because it adopts a soft margin over a hard margin. This means that it allows for *some* misclassification as a trade-off for achieving better generality. In the case of the data above, a hard margin is not the best because it is not easy to have a clear split between the data points as there are quite a few outliers near the margins. A hard margin would cause overfitting since it would work well for the test data but only consider the support vectors, which will not pick up on the general trend of the data, causing it to perform worse on future data. Meanwhile, a soft margin would apply a penalty for misclassified data, but would better suit future data.

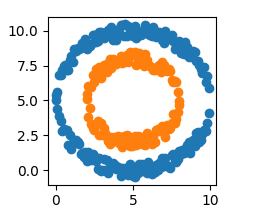
**Part B – Transforming Data**

The data in the first figure on the right has been gathered based on one property and its corresponding classification, e.g. the data point where x = 3 is classified as red, and the data point where x = 8 is classified as blue.

Explain briefly how the data has been transformed to produce the figure below it, and why this is useful.

The data has been transformed by a polynomial kernel, meaning that a transformation of has taken place. In this case, the data is not linearly separable, as it is impossible to specify one point (hyperplane) that accurately splits all of the data (one point is considered because the data is 1-dimensional, so a 0-dimension hyperplane needs to be used). The point of the kernel function is to apply a transformation to map the data to a higher dimension (2 dimensions in this case) so that we can apply a linear hyperplane (a 1-dimensional line). As is visible above, this data can now be easily separated by a hyperplane and can then be remapped to the original 1-dimensional data.

**Part C – Beyond 3 Dimensions**



Explain briefly how an SVM separates these data, comparing the technique to the examples above.

A similar kernel trick can be applied to this data. In this case, the kernel transformation would map the data to 3 dimensions so that it can be separated by a 2-dimensional hyperplane since the current data cannot be separated. A possible transformation could be where and are the current horizontal and vertical axes respectively and is the new axis. As such, an SVM can now create a hyperplane that will then be mapped back down to 2 dimensions.