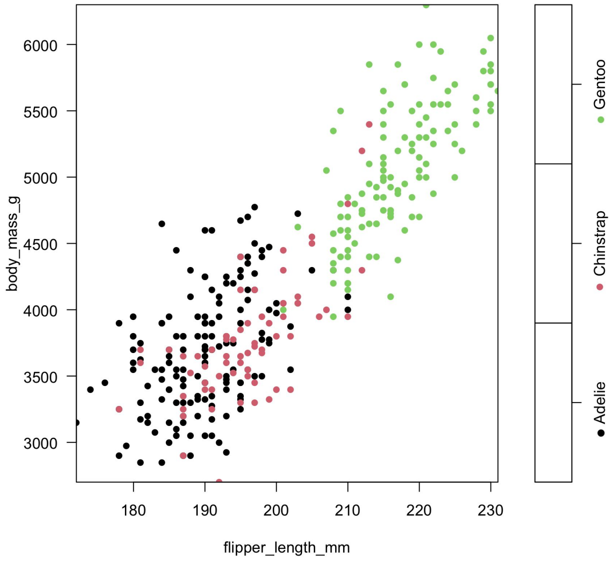
Support Vector Machines

Support Vector Machines (SVMs)

Supervised learning model for classification or regression

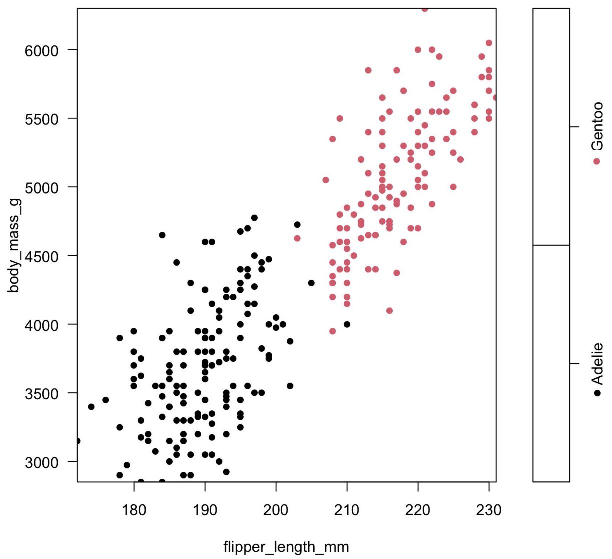
- Works well for finding the boundary between classes
- Efficient by reducing the the data set
- Supporting vectors
- Kernel functions to determine the shape of the boundary

Training Dataset - Penguins

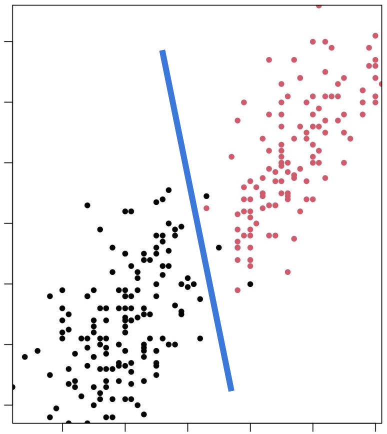


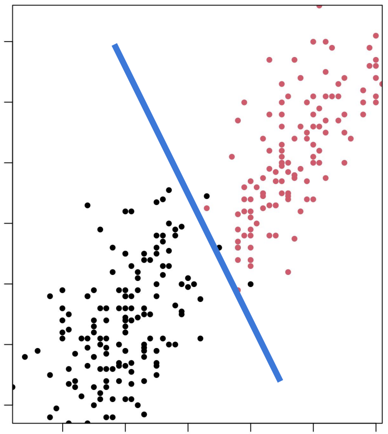
From the Palmer Penguins Library

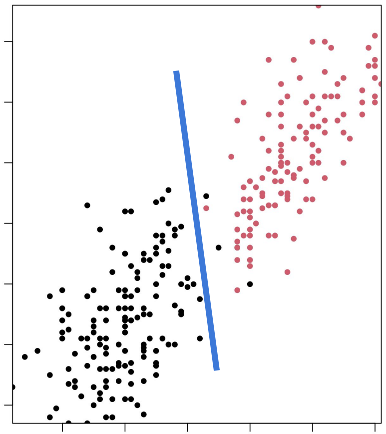
Classify Penguins Between Gentoo and Adelie



Where To Divide?







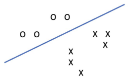
SVM Process

1. Choose a general shape for the decision body (linear, polynomial, radial)
2. SVM transforms the data into a higher dimensional space
3. SVM finds a class boundary (hyperplane)
4. SVM projects the class boundary back into the original dimensions space
5. SVM determines its most important data points
6. We have model (a class boundary and SV’s) for classification

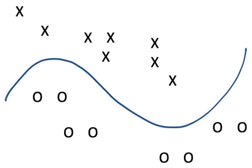
Step 1 - Choosing a Decision Boundary Type

SVM’s can create boundaries bound by various expressions:

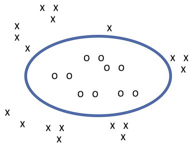
Linear



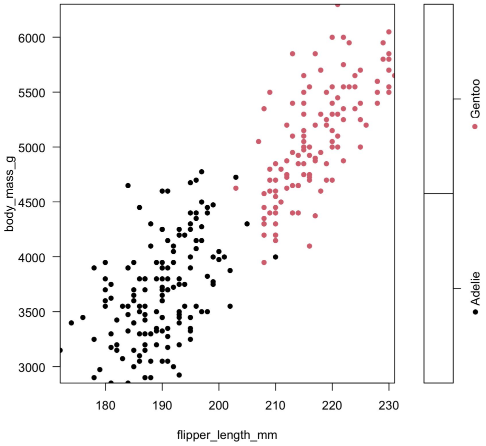
Polynomial



Radial Basis



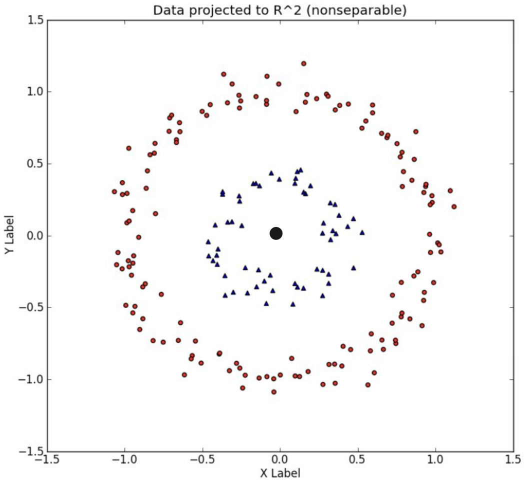
Step 2 - Transforming Linearly Separable Data



Do we need another dimension to separate the classes?

No. Data that is separable in 2D will also be separable in 3D

Step 2 - Transforming Non Linear Data



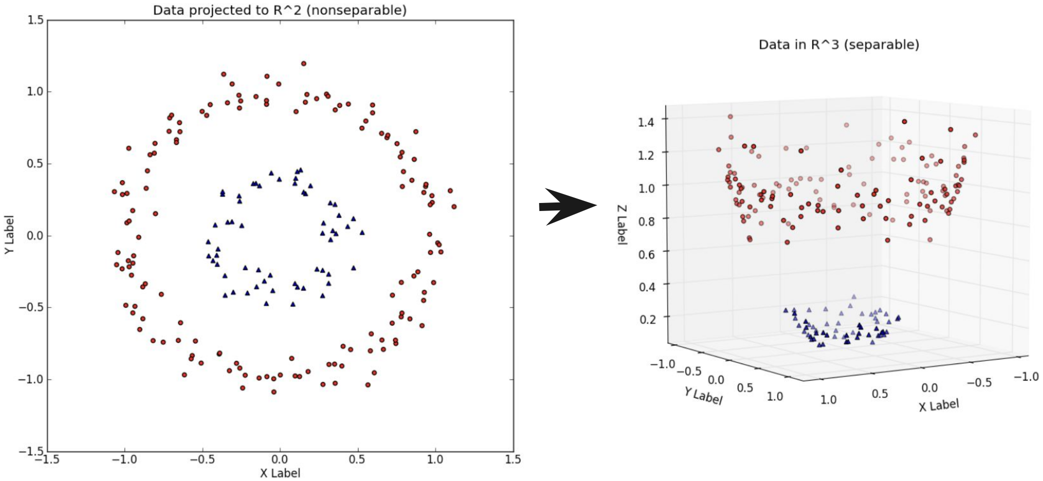
In 2D, no line can separate the two classes.

However for a z coordinate, we can define for each point:

z = distance from the origin = sqrt(x^2 + y^2)

http://www.eric-kim.net/eric-kim-net/posts/1/kernel_trick.html

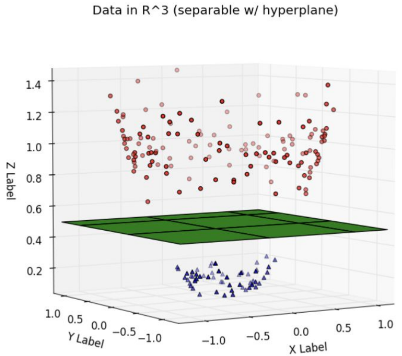
Step 2 - Transforming Non Linear Data

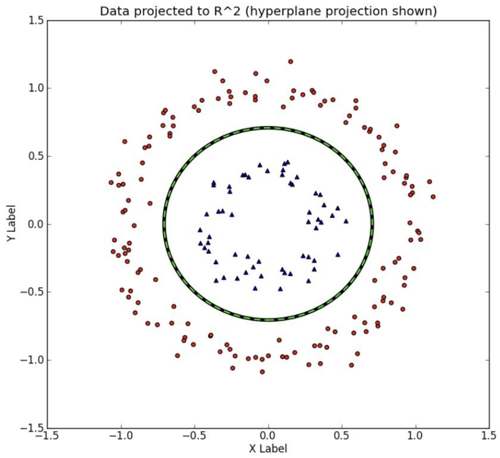


Transforming into 3D, a separation appears

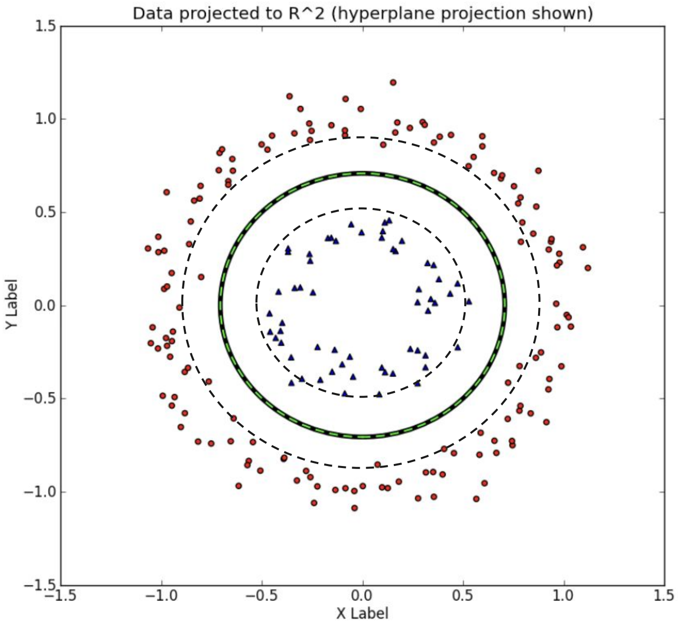
http://www.eric-kim.net/eric-kim-net/posts/1/kernel_trick.html

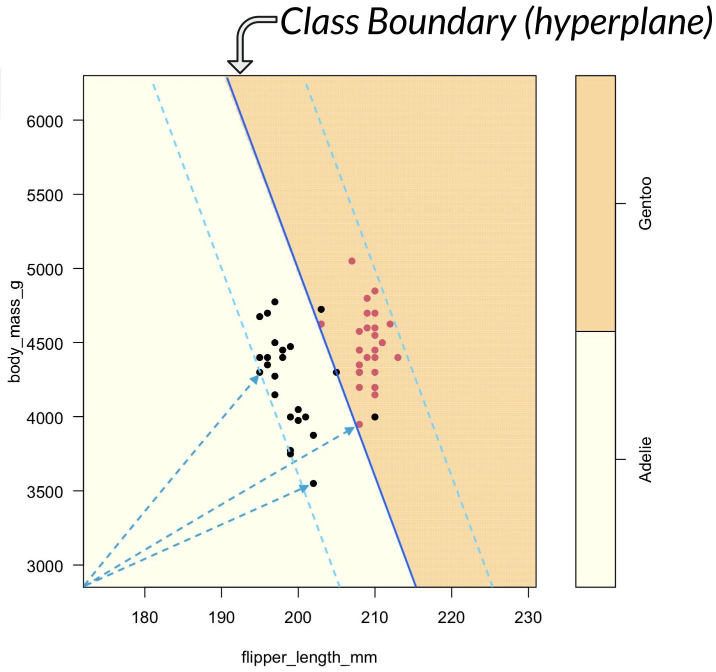
Step 3 & 4 - Class Boundaries and Back





Step 5 - Model with Margins



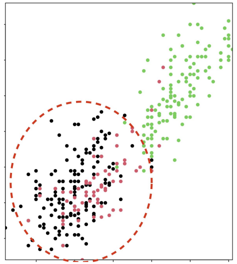


Efficiencies

- Reduced dataset
- Skip step 2 - transformation (kernel trick)
- Comparing/prediction through dot-products of vectors
- Works with datasets with many features (ncol >> nrows)

But!

- Doesn’t work well if the classes overlap a lot
- High number of SVs makes running slow



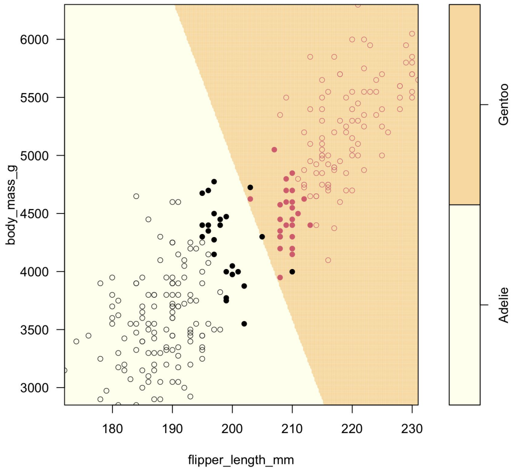
Cross Validation

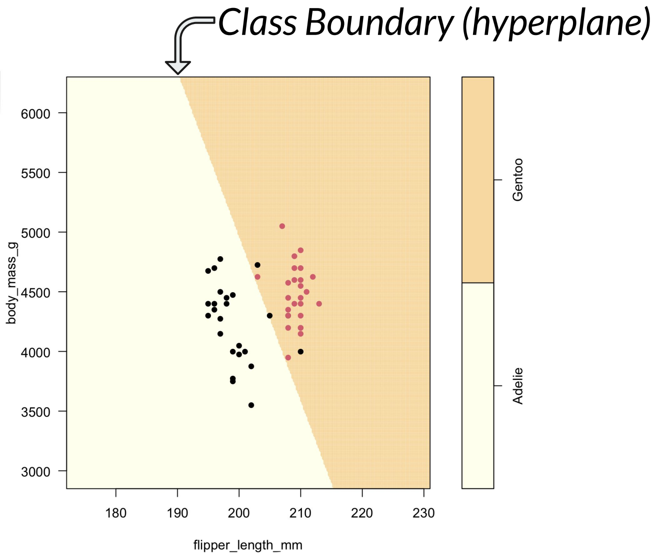
Linear Kernel : $x_i \bullet x_j$ -> tune (cost)

Polynomial Kernel: $(x_i \bullet x_j + coef)^d$ -> tune(cost, degree, coef)

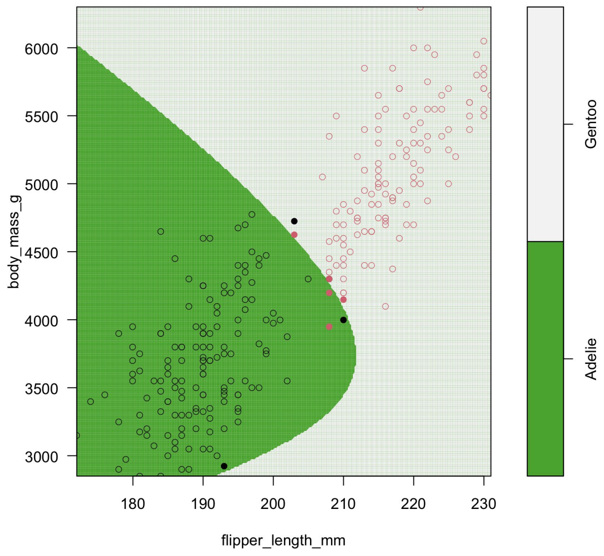
Radial Basis Kernel: $exp^{(-\gamma \mid xi - xj \mid \hat{}2)}$ -> tune(cost, gamma)

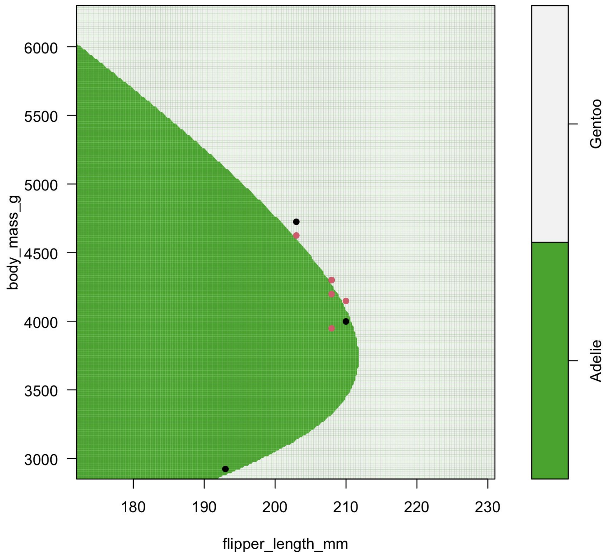
Tuned Penguin Model with Linear Kernel



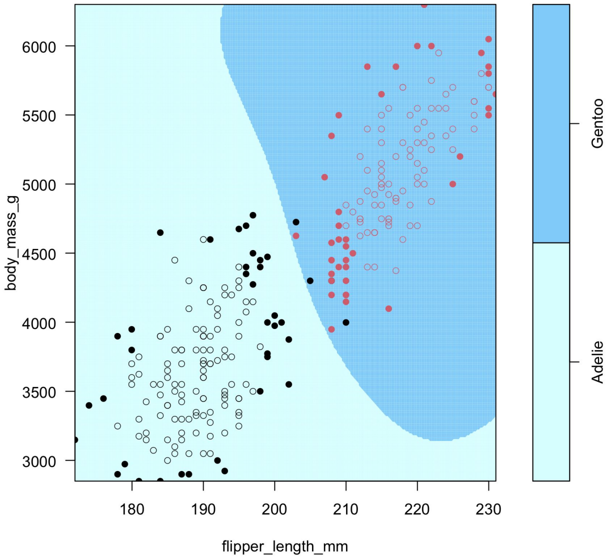


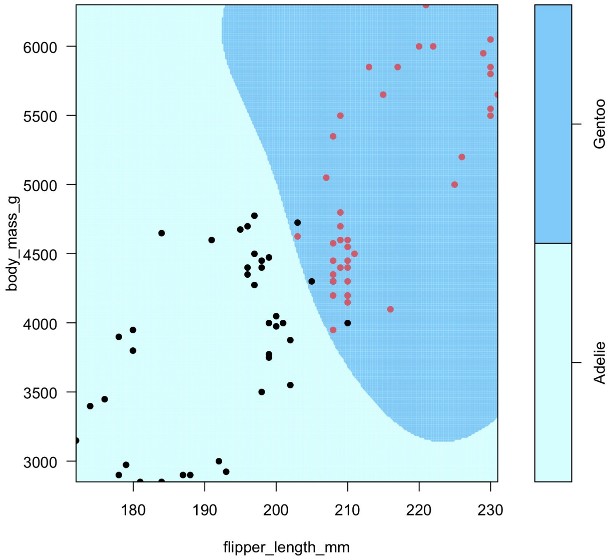
Tuned Penguin Model with Polynomial Kernel





Tuned Penguin Model with Radial Basis Kernel





Applications of SVMs

- Face recognition
- SPAM detection
- Handwriting recognition
- Datasets where the number of dimensions is higher than the number samples
- NLP