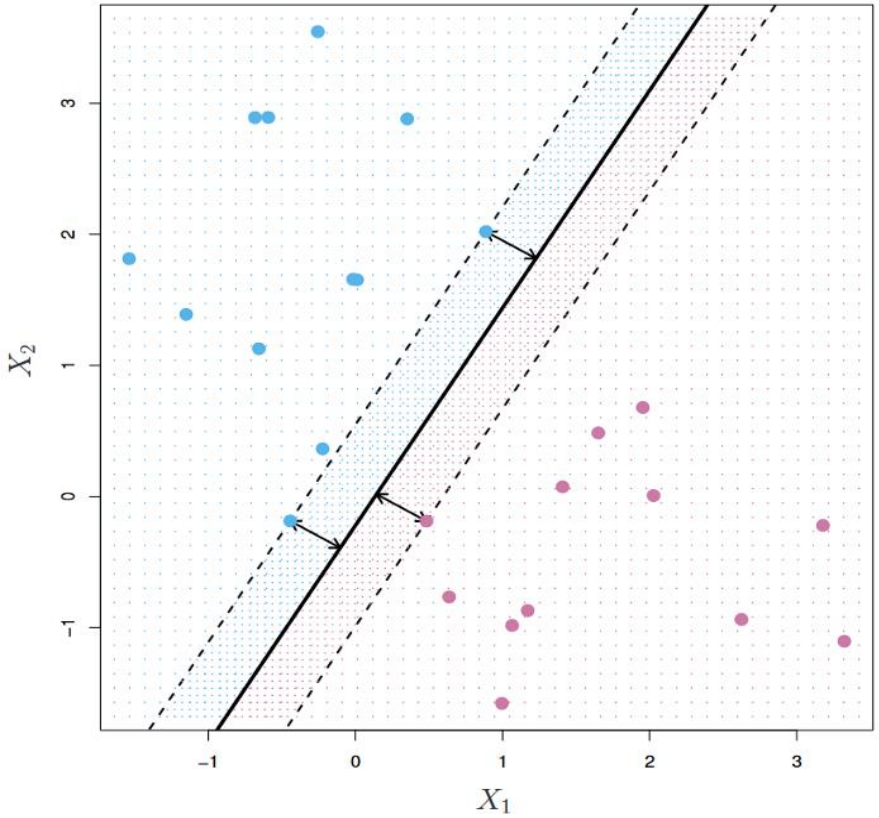
Support Vector Machine

Lec Shahriar Rahman Khan
Dept of CSE, MIST

Understanding SVM with a MEME



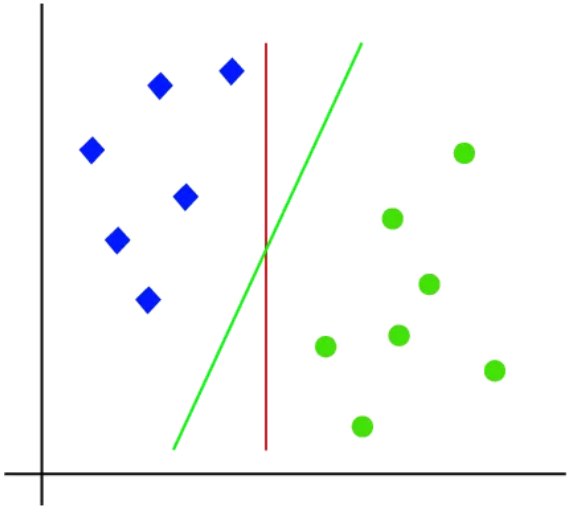
Introduction to ‘Support Vector Machine’

- *Support Vector Machine (SVM) is a relatively simple **Supervised** Machine Learning Algorithm used for classification and/or regression. It is more **preferred for classification** but is sometimes very useful for regression as well. SVM finds a **hyper-plane** that creates a boundary between the types of data. we use for both classification and regression.*

- *The goal of the SVM algorithm is to create the **best line** or **decision boundary** that can segregate n-dimensional space into classes so that we can easily put the new data point in the correct category in the future. This best decision boundary is called a **hyper-plane**.*

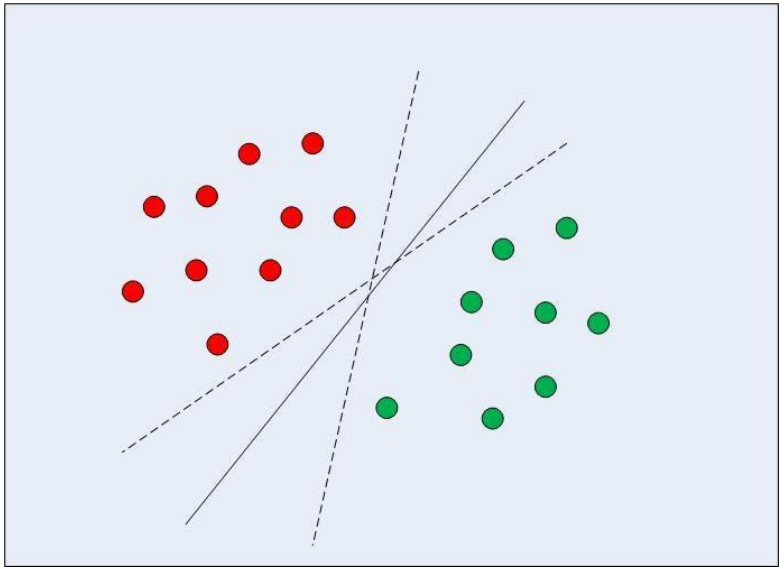
Introduction to ‘Support Vector Machine’

- In two dimensional space, this optimal hyper-plane can be thought of as a line dividing the space into two parts : where one part of the space contains data points which belong to one class while other part of the space contains data points which belong to the other class. The concept of lines acting as a classifier is only true if the data points are linearly separable.

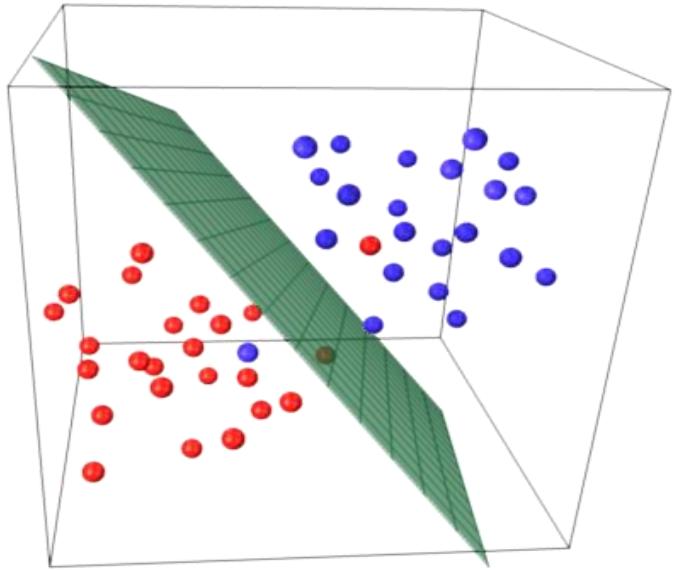


What is Hyperplane ?

- A hyperplane is plane of *n-1* dimensions in *n* dimensional feature space, that separates the two classes. For a 2-D feature space, it would be a line and for a 3-D Feature space it would be plane and so on.



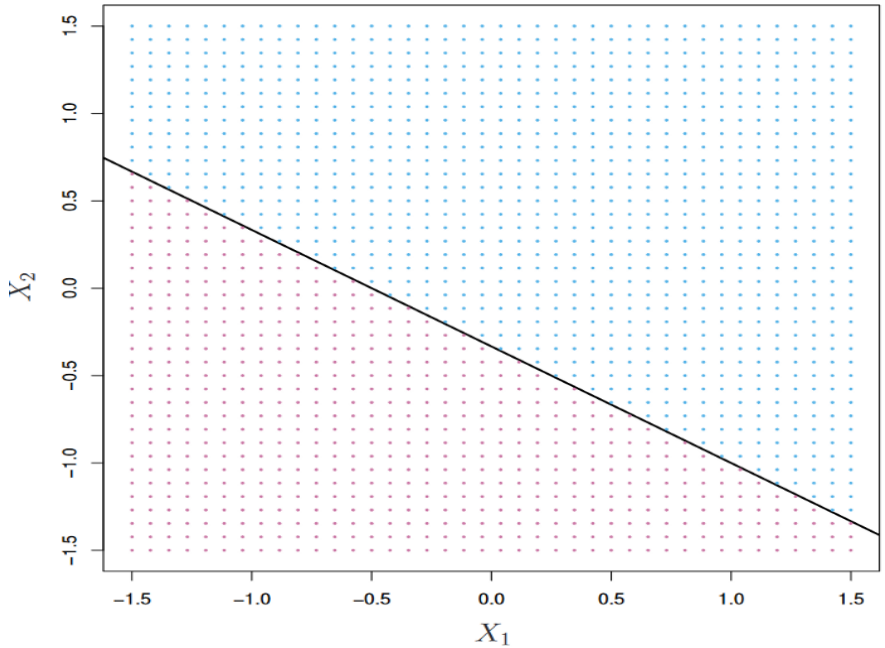
2D space



3D space

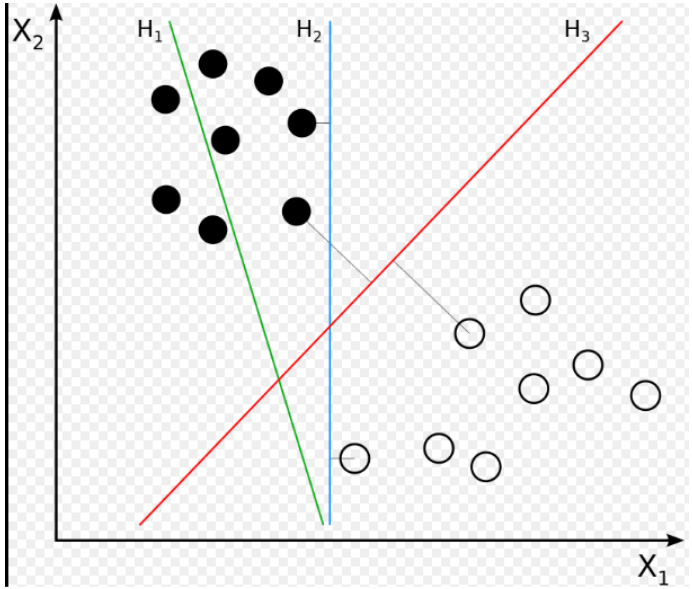
What is Hyperplane ?

- Mathematically, the hyperplane is simply: $\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p = 0$
- If *X* satisfies the equation above, then the point lies on the plane. Otherwise, it must be on one side of the plane as shown below.



Which Hyperplane to Choose?

- In general, if the data can be perfectly separated using a hyperplane, then there is an infinite number of hyperplanes (a great problem), since they can be shifted up or down, or slightly rotated without coming into contact with an observation. Our target: **Which straight line is the best for the classification task ?**



H1 does not separate the classes. H2 does, but only with a small margin. H3 separates them with the maximal margin

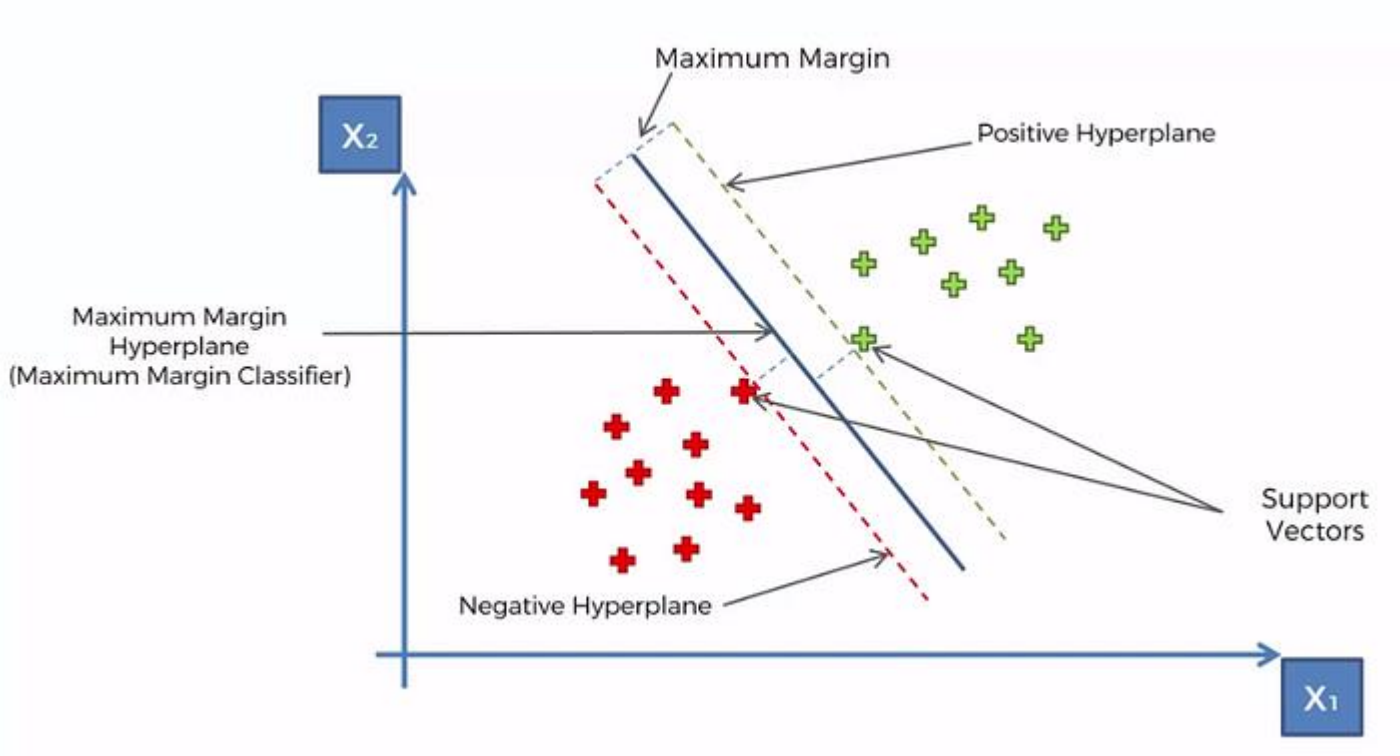
Optimal Hyperplane

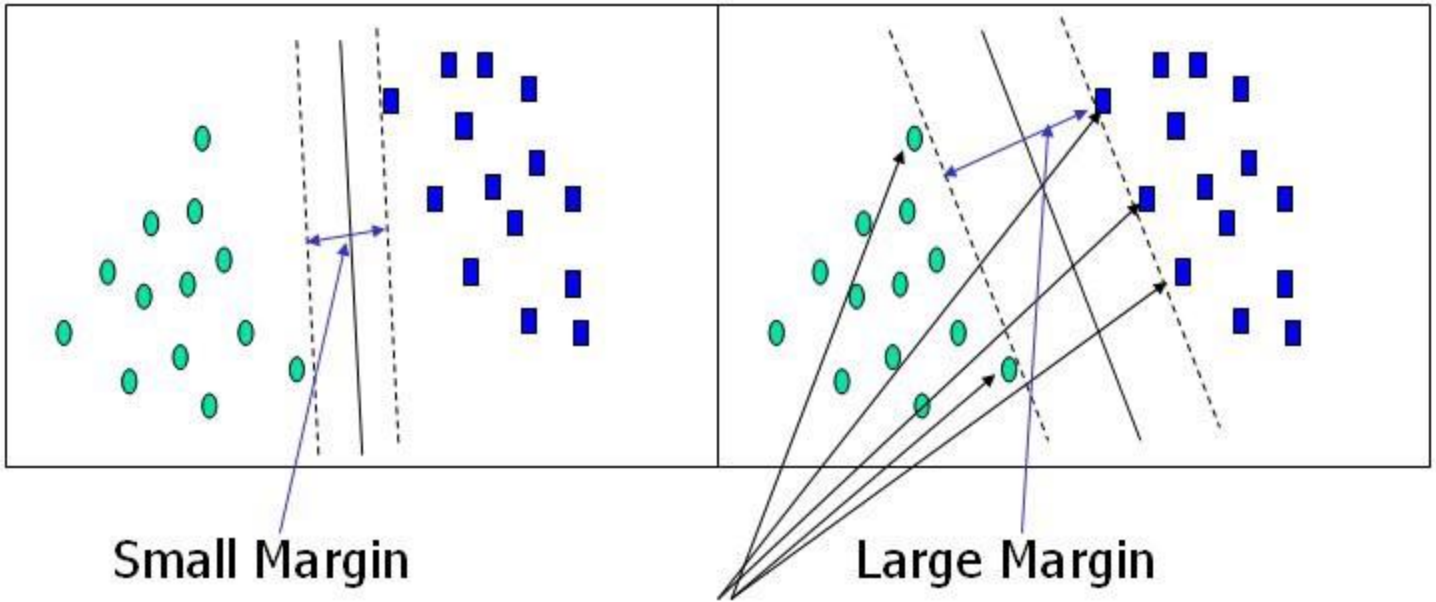
- In previous sections we have been using the term **“optimal hyper-plane”.** In this section we will be explaining what do we mean by **“optimal hyper-plane”.**

- By optimal hyper-plane we are referring to that separating hyper-plane from which the data points of classes are at farthest distance away on either side .

- The optimal hyper-plane is defined by the plane that maximizes the perpendicular distance between the hyper-plane and the closest samples. This perpendicular distance can be spanned with **support vectors**.

- Now we will be introducing a term called **“margin”**. Let us consider the figure in next slide.


Maximum Margin
X2
Positive Hyperplane
Maximum Margin
Hyperplane
(Maximum Margin Classifier)
Support
Vectors
Negative Hyperplane
X1

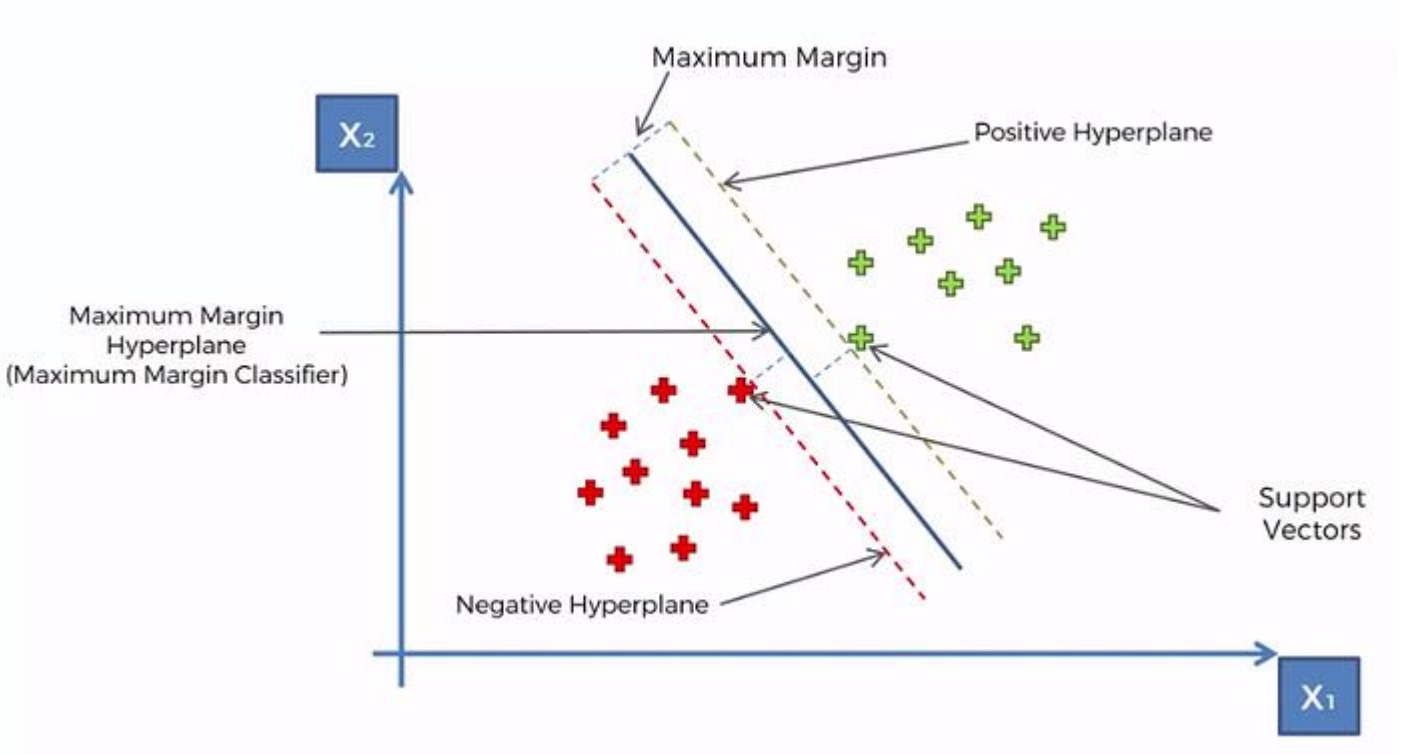

Small Margin
Large Margin
Support Vectors

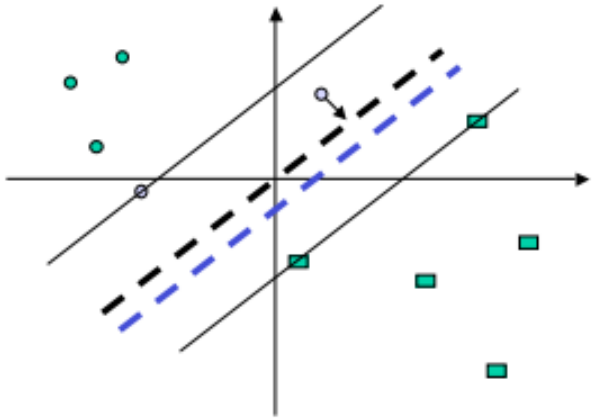
What is Support Vectors?

- We can see that there are three hyper-planes:
  - The hyper-plane touching the points of the positive class = **positive hyper-plane**.
  - The hyper-plane touching the points of the negative class = **negative hyper-plane**.
  - The hyper-plane situated in between the positive and negative class = **separating hyper-plane**.
- All these three hyper-planes are parallel to each other.
- The distance between the positive and negative hyper-plane = "**margin**".
- maximize the margin, the accuracy of the classification task increases. **The wider the margin, the better it is for the classification task.**
- SVMs try to find a hyper-plane, that maximizes the margin (Winston terminology: the 'street'). Hence the optimal or separating hyper-plane is also called **"margin-maximizing hyper-plane".**

What is Support Vectors?

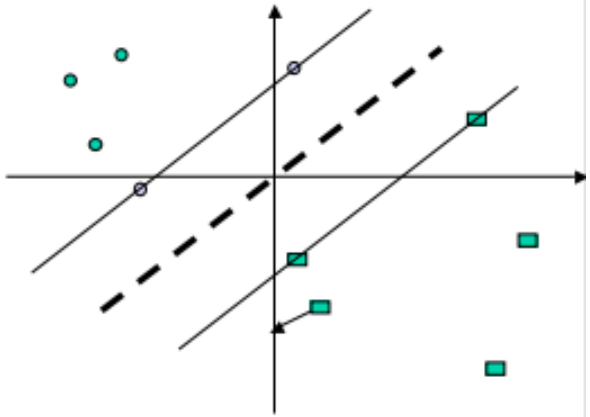
- **Support vectors** are data points that are closer to the hyperplane and influence the position and orientation of the hyperplane. Using these support vectors, we maximize the margin of the classifier.
- Support Vectors are those data points that touch the positive and negative hyper-planes. In the next Figure (slide 14), we can see that there are some data points which first touch the positive and negative hyper-planes, these data points are known as **“Support Vectors”**.
- Moving a support vector moves the decision boundary, but Moving the other vectors has no effect (see slide 15). Deleting the support vectors will change the position of the hyperplane. These are the points that help us build our SVM.


Maximum Margin
Positive Hyperplane
X2
Maximum Margin
Hyperplane
(Maximum Margin Classifier)
Support
Vectors
Negative Hyperplane
X1



Moving a support vector moves the decision boundary

Moving the other vectors has no effect



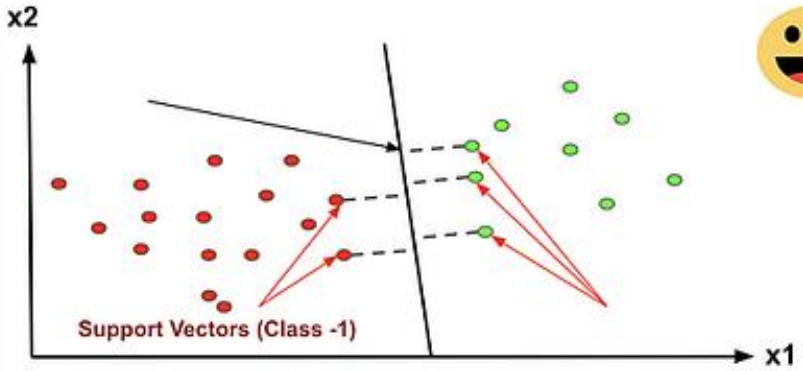
The optimization algorithm to generate the weights proceeds in such a way that only the support vectors determine the weights and thus the boundary

Understanding SVM with a MEME

Types of Support Vector Machine

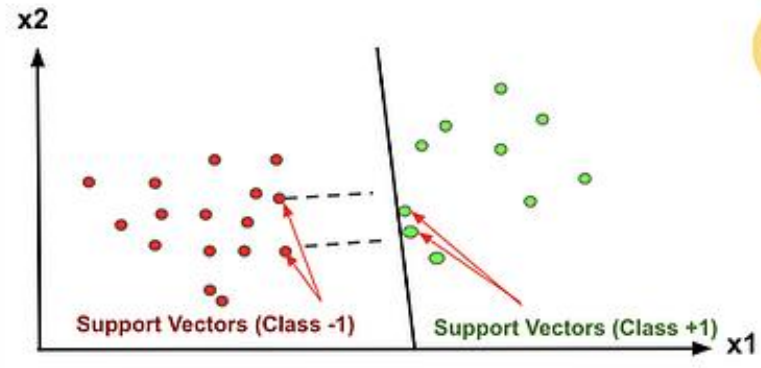
- There are two types of SVM
  - Hard SVM
  - Soft SVM
- **Hard SVM:**
  - Assuming that no points will lie on margin distance
  - There are no incorrectly classified points.
  - positive hyperplane is parallel to the main hyper plan and touches the positive support-vector.
  - The negative hyperplane is parallel to the main hyper plan and touches the negative support-vector.
  - The distance between a positive hyper plan and the negative hyper plan is called margin distance.

Good Margin VS Bad Margin



Good Margin

- all sector vectors have the same distance with the maximum margin hyperplane



Bad Margin

- very close to either class -1 support vectors or class +1 support vectors

Equation of Hyperplane in the ‘M’ dimension

$$y = w_0 + w_1 x_1 + w_2 x_2 + w_3 x_3 \ldots .$$
$$= w_0 + \sum_{i=1}^{m} w_i x_i$$
$$= w_0 + w^T X$$
$$= b + w^T X$$

where,

- Wi = vectors(W0,W1,W2,W3……Wm)
- Wo= constant
- X = variables/input vectors.
- **Distance between two hyperplanes is, d= 2/||w||. But is this calculated?**
- d will not be constant, it will be variable

Hard SVM

- The linearly separable problem, has two classes in it classified by the Positive class and negative class.
- If you guys observe there are three equations. The negative class has the equation **W^T*X+b=-1** because whenever a data point falls in that region the dot product of the vectors must be negative for the negative class.
- The Decision boundary has the equation **W^T*X+b=0** because whenever a data point falls on the decision boundary the dot product of the vectors must be equal to zero since it is in the middle of the both positive class and positive class.
- The positive class has the equation **W^T*X+b=+1** because whenever a data point falls in the positive class the dot product of the vectors must be positive.

Hard SVM

Define the hyperplanes H such that:

$w \bullet x_i + b \geq +1$ when $y_i = +1$

$w \bullet x_i + b \leq -1$ when $y_i = -1$

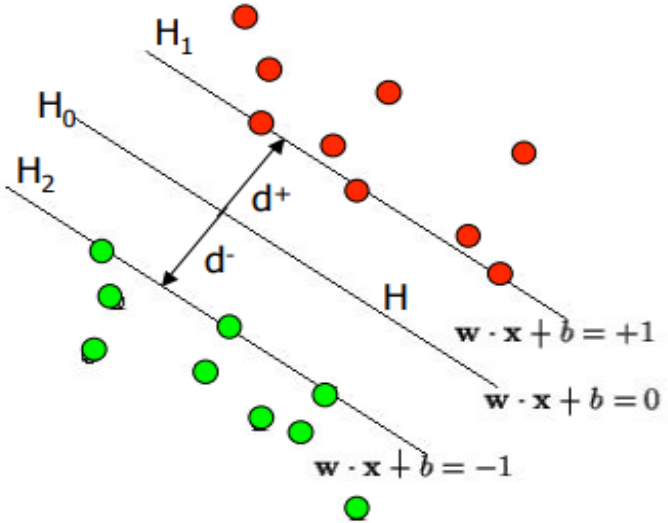
H_1 and H_2 are the planes:

H_1: $w \bullet x_i + b = +1$

H_2: $w \bullet x_i + b = -1$

The points on the planes H_1 and H_2 are the tips of the Support Vectors

The plane H_0 is the median in between, where $w \bullet x_i + b = 0$



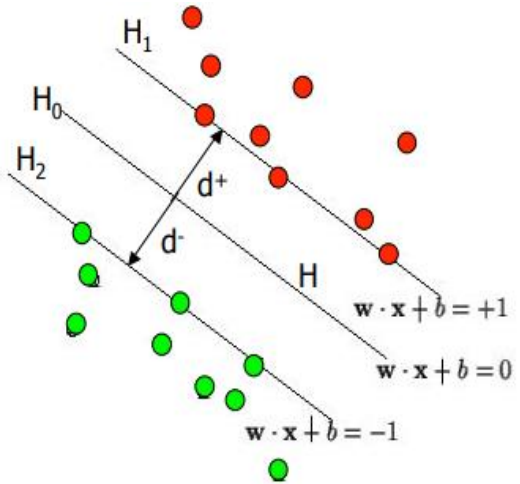
d+ = the shortest distance to the closest positive point

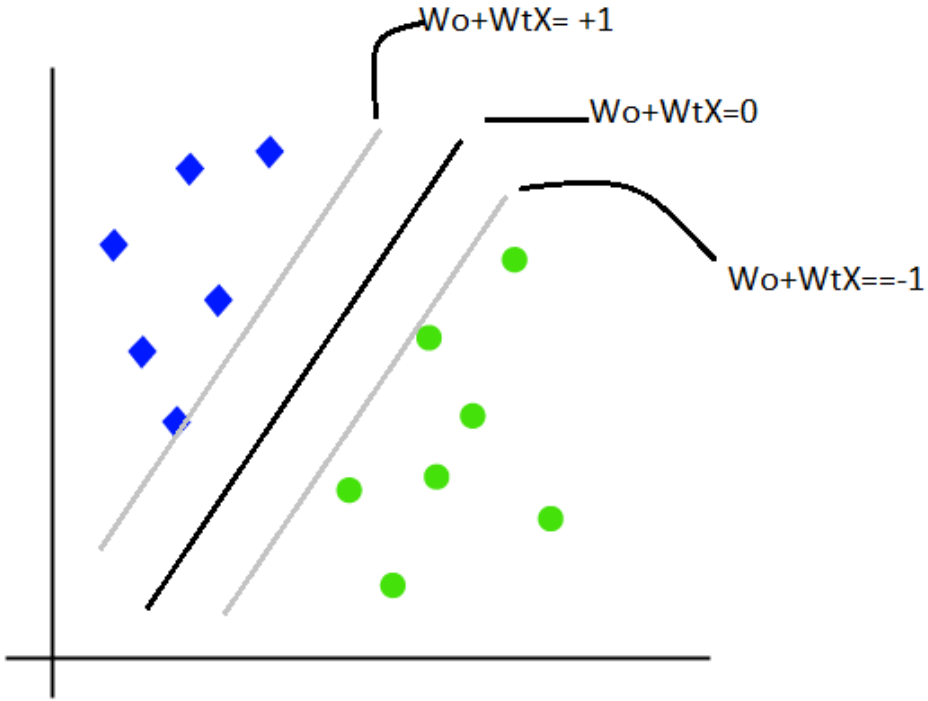
d- = the shortest distance to the closest negative point

The margin (gutter) of a separating hyperplane is d+ + d–.

Hard SVM

- Let, two points x1 and x2, where x1 is on negative plane and x2 is on positive plane. So,
- w^T•x1+b = -1
- w^T•x2+b = +1
- (x2-x1) w^T = 2
- (x2-x1) w^T/||w|| =2/||w||
- The total distance between H1 and H2 is thus: **2/||w||**
- **We need to maximize this by updating (w,b).**
  - **yi =+1 if xi•w^T+b ≥ +1**
  - **yi =–1 if xi•w^T+b ≤ –1**
- **Can be combined into: yi*(xi•w^T +b) ≥ 1**
- yi(W^T•xi + b) >+ 1 = for all correctly classified points.
- yi(W^T•xi +b) <- 1= for all incorrectly classified points.




Wo+WtX= +1
Wo+WtX=0
Wo+WtX==-1

Let the equation of the separating hyperplane be :

$$\pi : w^T.x + b = 0 \tag{1}$$

$$if \qquad \pi^+ : w^T.x + b = 1 \tag{2}$$

be the equation of the positive hyperplane

and

$$\pi^- : w^T.x + b = -1 \tag{3}$$

be the equation of the negative hyperplane

$$then\ margin = \frac{2}{||w||} \tag{4}$$

For SVMs we can write the constraint optimization problem as :

$$w^*, b^* = argmax_{w,b} \frac{2}{||w||} \tag{5}$$

$$such\ that\ \forall i,\ y_i\left(w^T x_i + b\right) \geqq 1$$

Applicable if data is linearly separable , all (+)ve pts lie on one side of π and all (−)ve pts lie on other side of π and there are no datapoints in between the π^+ and π^-.

Equal to 1 is for the Support Vectors.

Modification to Hard SVM to get Soft SVM

Let the equation of the separating hyperplane be :

$$\pi : w^T.x + b = 0 \tag{1}$$

$$if \quad \pi^+ : w^T.x + b = 1 \tag{2}$$

be the equation of the positive hyperplane

and

$$\pi^- : w^T.x + b = -1 \tag{3}$$

be the equation of the negative hyperplane

$$then\ margin = \frac{2}{\|w\|} \tag{4}$$

For SVMs we can write the constraint optimization problem as :

$$w^*, b^* = argmin_{w,b} \frac{\|w\|}{2} + C.\frac{1}{n}\sum_{i=1}^{n}\xi_i \tag{5}$$

$$such\ that\ \forall i,\ y_i\left(w^T x_i + b\right) \geqq 1 - \xi_i,\ \xi_i \geqq 0$$

ξ = *some units of distance away from the correct hyperplane in the incorrect direction.*

For correctly classified points $\xi_i = 0$ *i.e if* $y_i\left(w^T x_i + b\right) \geqq 1$

For incorrectly classified points $\xi_i > 0$

Equal to 1 *is for the Support Vectors.*

Soft SVM

- In equation 5, the first portion of the equation before the '+' sign is referred to as the **'regularization'** and the second portion is referred to as the **'Hinge Loss'**.
- 'C' is the hyper-parameter which is always a positive value.
- If 'C' increases, then overfitting increases and if 'C' decreases, then underfitting increases.
- For large values of 'C', the optimization will choose a smaller-margin hyper-plane if that hyper-plane does a better job of getting all the training points classified correctly.
- Conversely, a very small value of 'C' will cause the optimizer to look for a larger-margin separating hyper-plane, even if that hyper-plane misclassifies more points.

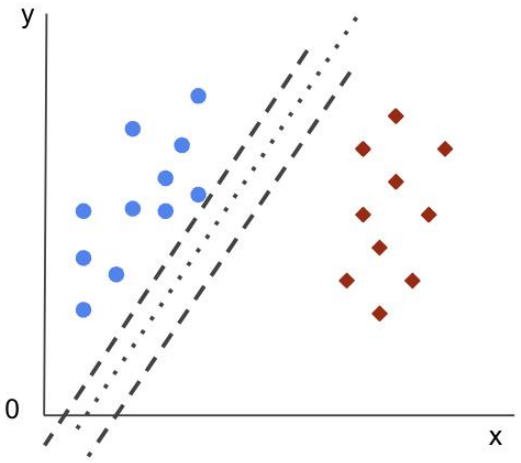
Soft SVM

- **When *C* is Small**:
  - A small value of *C* implies a high degree of regularization. The model is penalized heavily for misclassifying data points. In practical terms, this means that the SVM is willing to accept a larger margin (i.e., allow more training points to be inside the margin or even on the wrong side of the decision boundary) in exchange for better overall classification of the training data.
  - The SVM tries to find a larger-margin hyperplane, even if it doesn't classify all training points correctly. This is often referred to as a "soft margin.“
- **When *C* is Large**:
  - A large value of *C* implies lower regularization. The model is less tolerant of misclassifications and focuses more on getting every individual point correct. In this case, the SVM aims to classify all training points correctly, even if it means having a smaller margin or potentially a hyperplane that is not as well-separated from the classes.

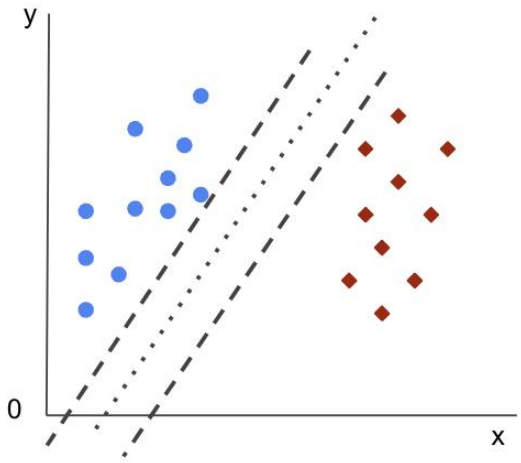
In summary, the choice of *C* balances the trade-off between maximizing the margin and minimizing the classification error.

Soft SVM

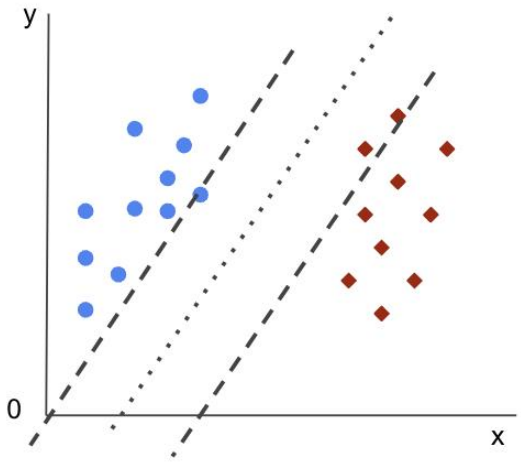
C = 100



C = 10

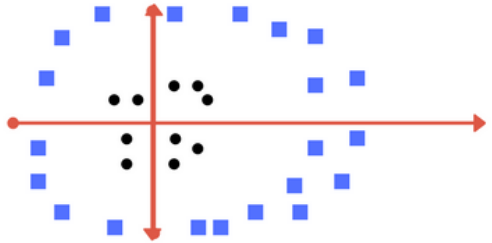


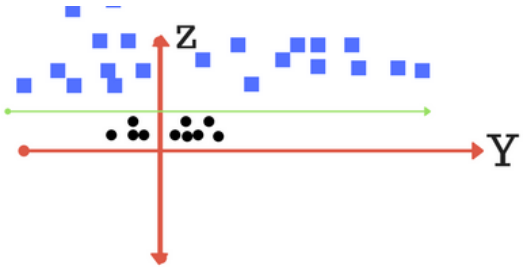
C = 1



SVM Kernel Trick

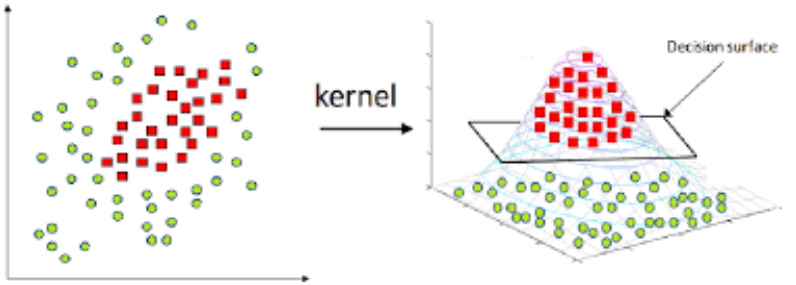
- Can it solve non-linear classification problem?
- Now consider what if we had data as shown in first image? Clearly, there is no line that can separate the two classes in this x-y plane. So what do we do? We apply transformation and add one more dimension as we call it z-axis. Lets assume value of points on z plane, $w = x^2 + y^2$. In this case we can manipulate it as distance of point from z-origin. Now if we plot in z-axis, a clear separation is visible and a line can be drawn .

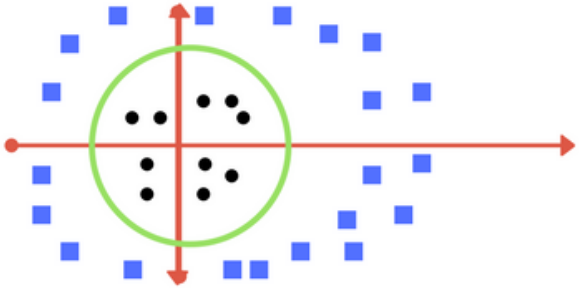




SVM Kernel Trick

- For the above example, the kernel will internally and implicitly transform the data which is 2 D to another higher dimensional space where the data will become linearly separable.
- What kernalization does is that it takes data which is ‘d’ dimensional and it does a feature transform internally and implicitly using the kernel trick to a dimension ‘d1’ , typically where d1 >d. In ‘d1’ , the data becomes linearly separable.
- By using the **RBF kernel**, the SVM projects the data into a higher-dimensional space where it can find a hyperplane that separates the classes.





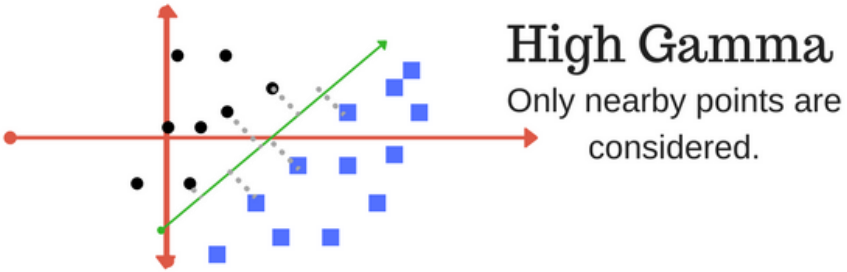
SVM Kernel Trick

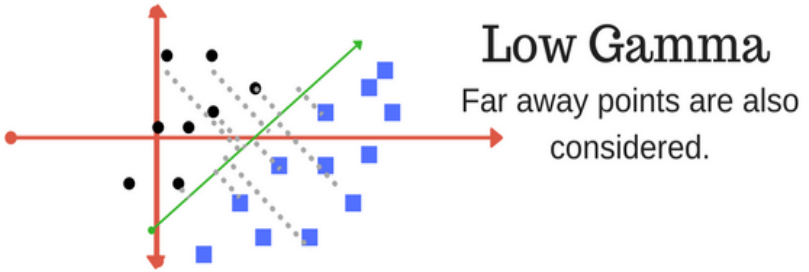
- We have to choose the correct kernel based on our application. There are various types of kernels :

1. *Polynomial Kernel (Homogeneous)* : $K(x_i, x_j) = \left(x_i^T x_j\right)^d$
2. *Polynomial Kernel (Inhomogeneous)* : $K(x_i, x_j) = \left(1 + x_i^T x_j\right)^d$
3. *Radial Basis Function Kernel* : $K(x_i, x_j) = exp\left(-\frac{\|x_i - x_j\|^2}{2\sigma^2}\right)$
4. *Domain Specific Kernel* : *Kernel chosen based on the domain of the application*

Hyper-parameter : Gamma

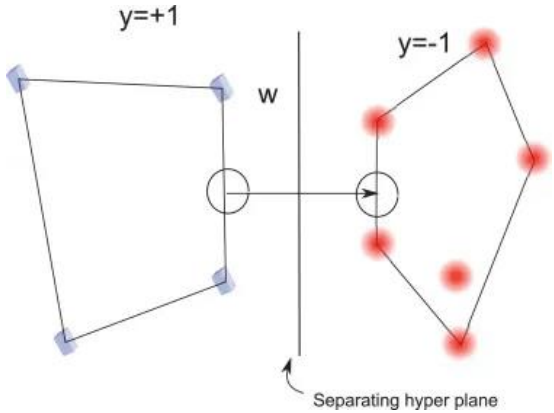
- There is a very important hyper-parameter in SVC called '**gamma**' which is used very often. Gamma is a hyperparameter used with non-linear SVM. One of the most commonly used non-linear kernels is the radial basis function (RBF)
- **Gamma :** The gamma parameter defines how far the influence of a single training example reaches, with low values meaning 'far' and high values meaning 'close'. In other words, with low gamma, points far away from plausible separation line are considered in calculation for the separation line. Where as high gamma means the points close to plausible line are considered in calculation.





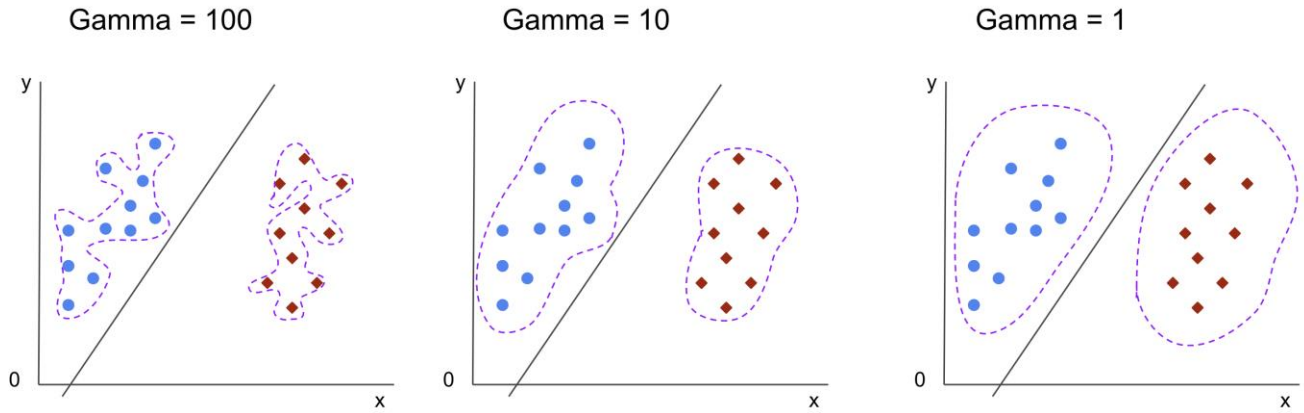
Decision Boundary and Region

- Decision Region:
  - A decision region is the set of points in the feature space that are classified into a particular class. It is the area of influence of a specific class in the feature space.
  - In a binary classification problem, there are two decision regions: one for each class. Each region consists of all the points for which the model predicts that class.
- Decision Boundary: The decision boundary is a hypersurface in the feature space that separates the different classes in a classification problem.



Effect of High and Low Gamma

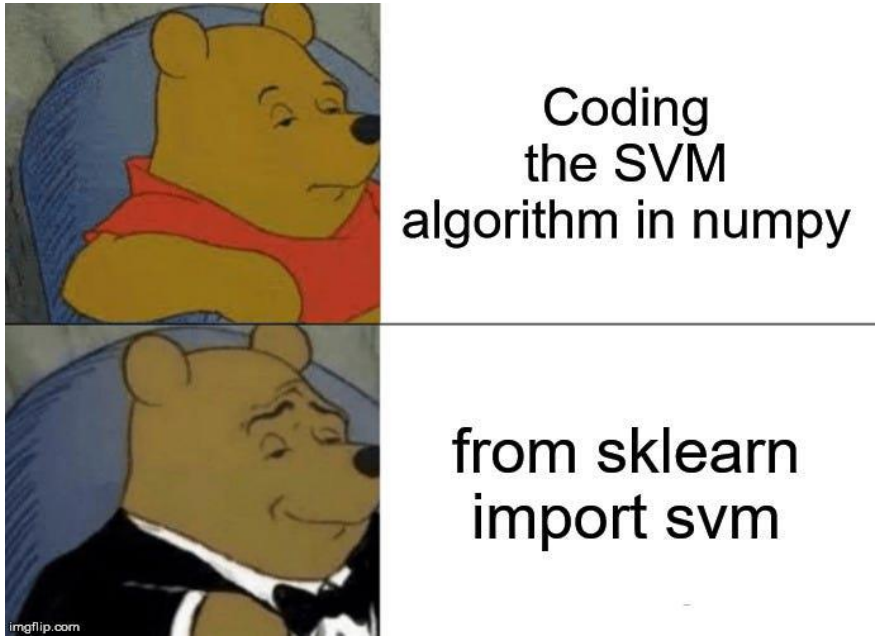
- In the low gamma case, the decision boundary is smoother and tries to find a balance between the classes. In the high gamma case, it closely follows the data points, possibly leading to overfitting.
- Choosing the right gamma is crucial. It's typically determined through techniques like cross-validation, where the dataset is split into training and validation sets, and different values of gamma are tried to see which one performs the best on the validation set.



Summary of Gamma and C

- However, it is only important to know that an SVC classifier using an RBF kernel has two parameters: '**gamma**' and '**C**'.
- **Observation : 'gamma'** is a parameter of the RBF kernel and can be thought of as the 'spread' of the kernel and therefore the decision region. When **'gamma'** is low, the 'curve' of the decision boundary is very low and thus the decision region is very broad. When **'gamma'** is high, the 'curve' of the decision boundary is high, which creates islands of decision-boundaries around data points.
- **'C'** is a parameter of the SVC learner and is the penalty for misclassifying a data point. When **'C'** is small, the classifier is okay with misclassified data points (high bias, low variance). When **'C'** is large, the classifier is heavily penalized for misclassified data and therefore bends over backwards avoid any misclassified data points (low bias, high variance).

Are We Going to Implement the Whole SVM?



Effect of High and Low Gamma

- Implementing SVM with Scikit-Learn
  - Dividing Data into Train/Test Sets
  - Training the Model
  - Making Predictions
  - Evaluating the Model
  - Interpreting Results

Thank You