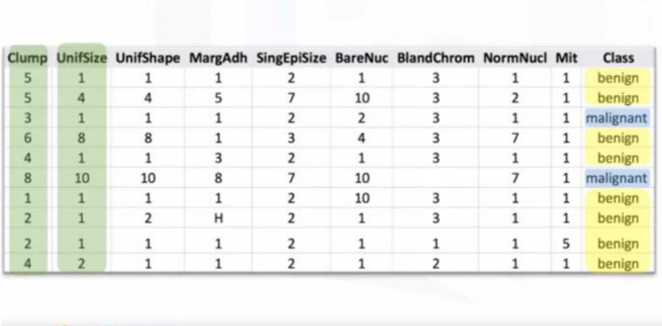
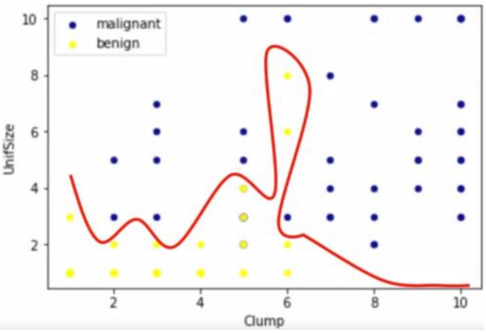
SUPPORT VECTOR MACHINE (SVM)

Support Vector Machine (SVM) can be describe as

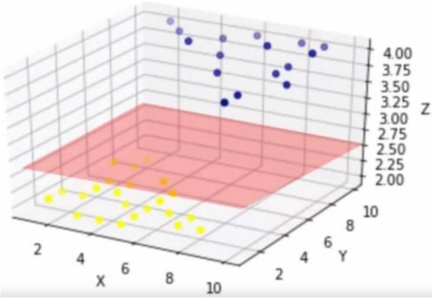
- A supervised algorithm that can classifies cases by finding a separator
  - SVM works by first mapping data to a high dimensional feature space so that data points can be categorized even when the data aren't otherwise linearly separable
  - then a separator is estimated for the data
- The data should be transformed in such a way that a separator could be drawn as a hyperplane
- Here is the example



Clump	UnifSize	UnifShape	MargAdh	SingEpiSize	BareNuc	BlandChrom	NormNucl	Mit	Class
5	1	1	1	2	1	3	1	1	benign
5	4	4	5	7	10	3	2	1	benign
3	1	1	1	2	2	3	1	1	malignant
6	8	8	1	3	4	3	7	1	benign
4	1	1	3	2	1	3	1	1	benign
8	10	10	8	7	10		7	1	malignant
1	1	1	1	2	10	3	1	1	benign
2	1	2	H	2	1	3	1	1	benign
2	1	1	1	2	1	1	1	5	benign
4	2	1	1	2	1	2	1	1	benign

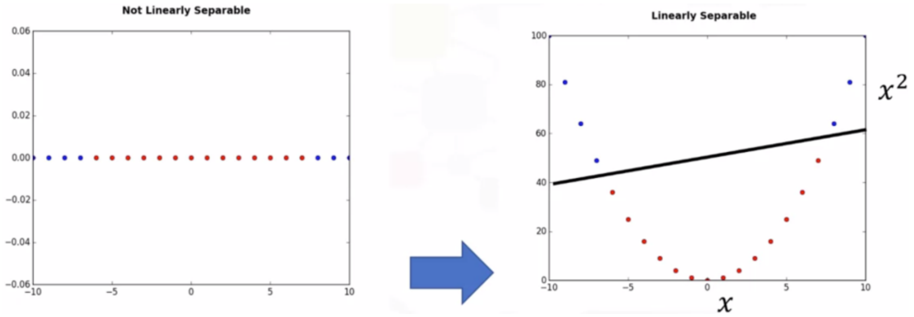


  - Consider the figure on the right which shows the distribution of a small set of cells only based on their *UnifSize* and *Clump* thickness
  - As we can see, the data points fall into 2 different categories – it represents a linearly non separable dataset
  - The 2 categories can be separated with a curve but not a line, that is it represents a linearly non separable dataset which is the case for most real-world data sets
  - We can transfer this data to a higher-dimensional space like the graph on the right which is mapping it to a 3-dimensional space
  - After the transformation, the boundary between 2-categories can be defined by a hyperplane
  - As we are now in 3-dimensional space, the separator is shown as a plan
  - This plane can be used to classify new or unknown cases



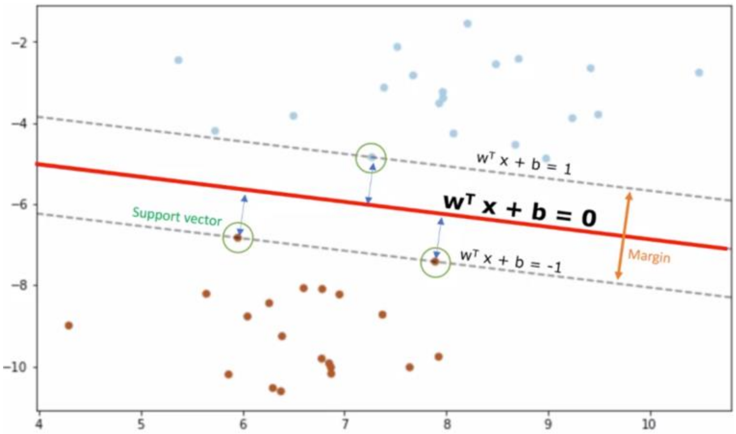
  - Therefore, the SVM algorithm output an optimal hyperplane that categorizes new examples

This is how we transfer our data in such a way that a separator could be drawn as a hyperplane:



- For the example the graph on the left, imagine that our dataset is 1-dimensional data means we have only 1 feature of X and it’s not linearly separable
- We transfer it into a 2-dimensional space, we can increase the dimension of data by mapping x into a new space using a function with outputs x and x^2 – $\emptyset(x) = [x, x^2]$
- Now the data is linearly separable like the graph on the right
- Notice that, as we are in 2-dimensional space the hyperplane is a line dividing a plane into 2 parts where each class lays on either side
- Now we can use the line to classify the new cases
- Basically, mapping data into a higher dimensional space is called **Kernelling**
- The mathematical function used for the transformation is known as the Kernel function and can be of different types
  - Linear
  - Polynomial
  - Radial Basis Function (RBF)
  - Sigmoid
- There is no easy way of knowing which function performs best with any given dataset, we usually choose different functions in turn and compared the result

And this is how we can find the best or optimized hyperplane separator after transformation:



- Basically, SVMs are based on the idea of finding a hyperplane that best divides a dataset into 2 classes
- As we're in a 2-dimensional space, we can think of hyperplane as a line that linearly separates the blue points from the red points
- One reasonable choice as the best hyperplane is the one that represents the largest separation or margin between the 2 classes
- So, the goal is to choose a hyperplane with as big a margin as possible
- Examples closest to the hyperplane are support vectors and it's intuitive that only support vectors matter for achieving our goal thus other trending examples can be ignored
- We try to find the hyperplane in such a way that is has the maximum distance to support vectors
- Please note that the hyperplane and boundary decision lines have their own equations
- So, finding the optimized hyperplane can be formalized using an equation which involves a bit more math
    - Find $\boldsymbol{w}$ and $\boldsymbol{b}$ such that $\Phi(w) = \frac{1}{2}w^T w$ is minimized
    - And for all $\{(x_i, y_i)\}: y_i(w^T x_i + b) \geq 1$
- That said the hyperplane is learned from training data using an optimization procedure that maximizes the margin
- Like many other problems, this optimization problem can also be solved by **Gradient Descent** therefore the output of the algorithm is the value of $\boldsymbol{w}^T\boldsymbol{x} + \boldsymbol{b}$ for the line and we can make classifications using this estimated line
- It's enough to plug in input values into the line equation then we can calculate whether an unknown point is above or below the line
- If the equation returns a value $> \mathbf{0}$ then the point belongs to the first class which is above the line, and vice-versa

Pros and Cons of SVM:

- **Advantages**
    - They're accurate in high-dimensional spaces
    - Memory efficient – the use a subset of training points in the decision function called support vectors
- **Disadvantages**
    - The algorithm id prone for over-fitting if the number of features is much greater than the number of samples
    - SVMs don't directly provide probability estimates which are desirable in most classification problems
    - SVMs aren't very efficient computationally if our dataset is very big – such as when we have more than 1000 rows

We should use SVM in situation of:

- Image recognition – SVM is good for image analysis tasks such as image classification and hand written digit recognition
- Text category assignment – SVM is very effective in text mining tasks, particularly due to its effectiveness in dealing with high-dimensional data
  - Detecting spam
  - Text category assignment
  - Sentiment analysis
- Gene Expression Data Classification – because of its power in high-dimensional data classification
- SVM can also be used for other types of ML problems
  - Regression
  - Outlier detection
  - Clustering