MINI PROJECT 2

Title: Support Vector Machine

Problem Statement: Apply Support Vector Machine on the dataset obtained from UCI ML repository.

Objective: Implementation of Support Vector Machine algorithm for classification.

Outcome: Classification of Mushroominto edible and poisonous.

Prerequisites: Python 3, Jupyter Notebook, Operating System: Ubuntu / Windows.

Theory:

Support Vector Machine

A Support Vector Machine (SVM) is a discriminative classifier formally defined by a separating hyperplane. In other words, given labeled training data (supervised learning), the algorithm outputs an optimal hyperplane which categorizes new examples. In two dimensional space this hyperplane is a line dividing a plane in two parts where in each class lay in either side.

I. Data is separated among in one quadrant.

Suppose you are given plot of two label classes on graph as shown in image (A). We have to decide the separating line for two classes.

Image A: Draw a line that separates black circles and blue squares.

something similar to the following image (image B). It fairly separates the two classes. Any point that is left of line falls into black circle class and on right falls into blue square class. **Separation of classes which is called as SVM.** It finds out a line/ hyper-plane (in multidimensional space that separate outs classes).

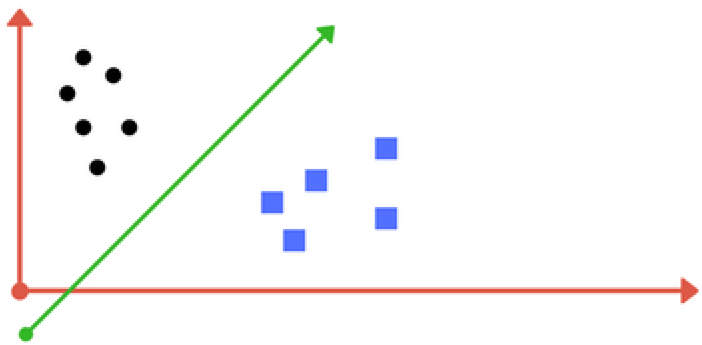


Image B: Sample cut to divide into two classes.

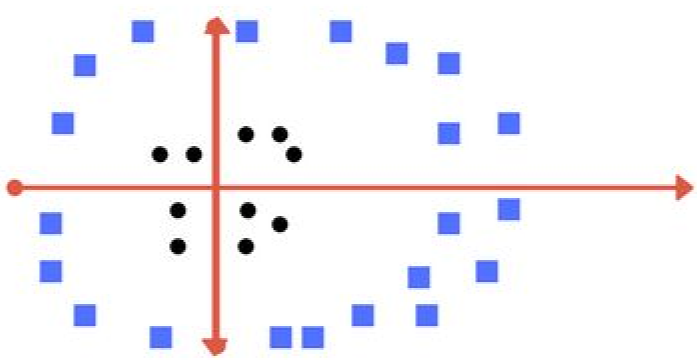
II. Data is separated across all the quadrants.

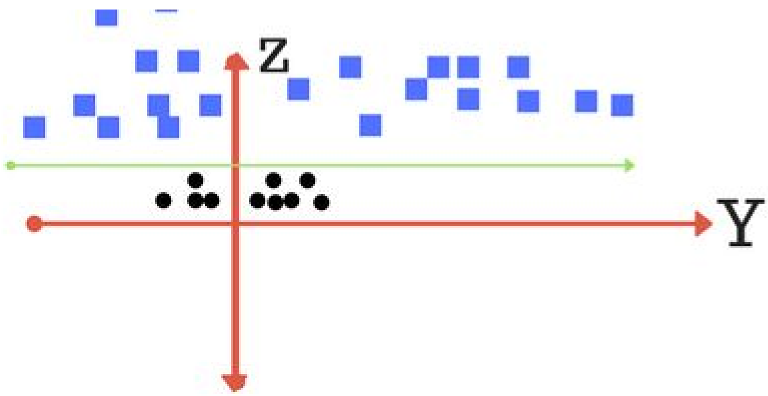
Consider data as shown in the image below.

1)There is no line that can separate the two classes in this x-y plane. We apply transformation and add one more dimension as we call it z-axis.

2) We assume value of points on the z plane, $w = x^2 + y^2$.

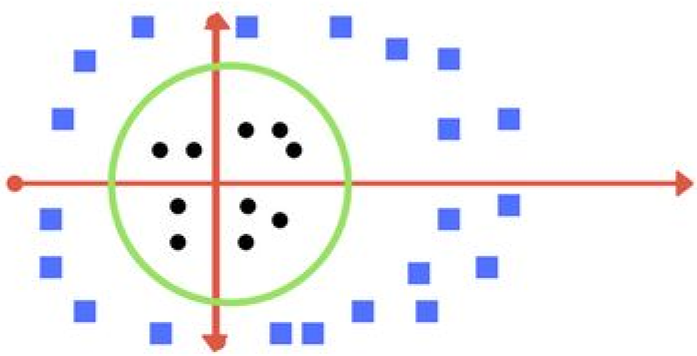
3) In this case we can manipulate it as distance of point from z-origin. Now if we plot in z-axis, a clear separation is visible and a line can be drawn .



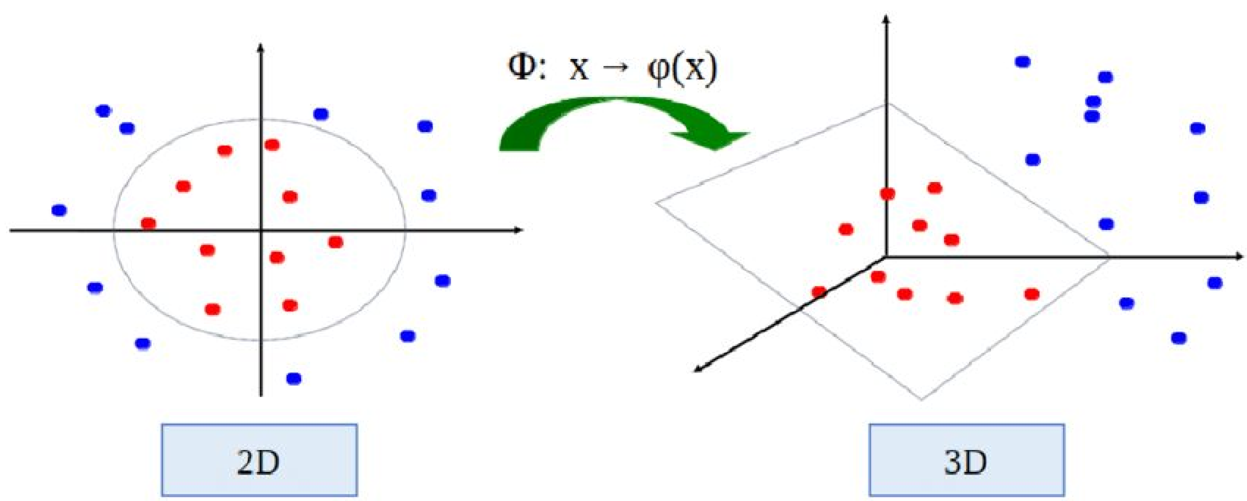


plot of z-y axis. A separation can be made here.

When we transform back this line to original plane, it maps to circular boundary as shown in image E. These transformations are called kernels



Transforming back to x-y plane, a line transforms to circle.



III. Data is Overlapping :

What if data plot overlaps? Or, what in case some of the black points are inside the blue ones? Which lie among 1 or 2?should we draw?

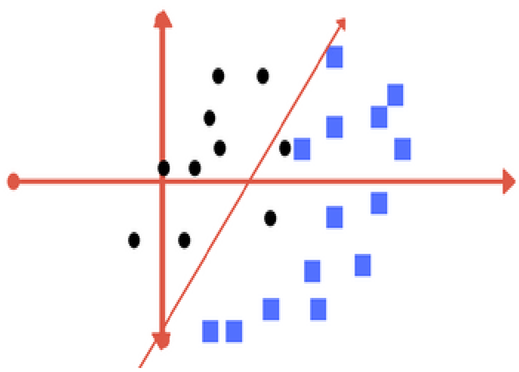


Image 1

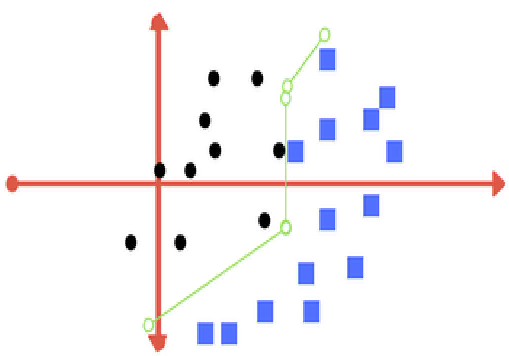
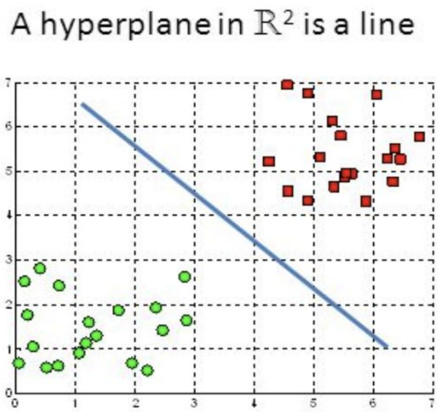


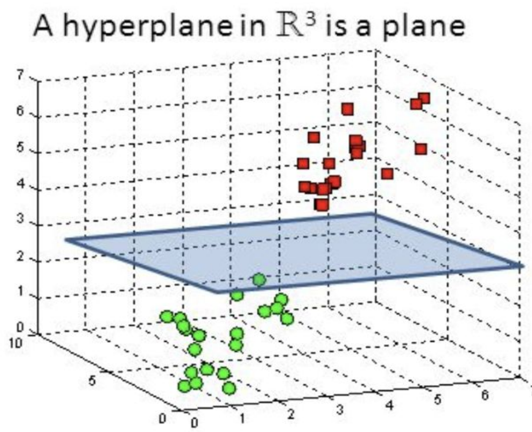
Image 2

1)The first one tolerates some outlier points.

2)The second one is trying to achieve 0 tolerance with perfect partition.

But, there is a trade off. In real world application, finding the perfect class for millions of training data set takes a lot of time. As you will see in coding. This is called regularization parameter. In the next section, we define two terms regularization parameter and gamma. These are tuning parameters in SVM classifier. Varying those we can achieve considerable non linear classification line with more accuracy in a reasonable amount of time. One more parameter is kernel. It defines whether we want a linear of linear separation.





DataSet

This dataset includes descriptions of hypothetical samples corresponding to 23 species of gilled mushrooms in the Agaricus and Lepiota Family Mushroom drawn from The Audubon Society Field Guide to North American Mushrooms (1981). Each species is identified as definitely edible, definitely poisonous, or of unknown edibility and not recommended. This latter class was combined with the poisonous one. The Guide clearly states that there is no simple rule for determining the edibility of a mushroom; no rule like "leaflets three, let it be'' for Poisonous Oak and Ivy.
Attribute Information: (classes: edible=e, poisonous=p)

- Cap-shape:

  bell=b,conical=c,convex=x,flat=f, knobbed=k,sunken=s

- Cap-surface:

  fibrous=f,grooves=g,scaly=y,smooth=s

- cap-color: brown=n,buff=b,cinnamon=c,gray=g,green=r,pink=p,purple=u,red=e,white=w,yellow=y
- bruises: bruises=t,no=f
- odor: almond=a,anise=l,creosote=c,fishy=y,foul=f,musty=m,none=n,pungent=p,spicy=s
- gill-attachment: attached=a,descending=d,free=f,notched=n
- gill-spacing: close=c,crowded=w,distant=d
- gill-size: broad=b,narrow=n
- gill-color: black=k,brown=n,buff=b,chocolate=h,gray=g, green=r,orange=o,pink=p,purple=u,red=e,white=w,yellow=y
- stalk-shape: enlarging=e,tapering=t
- stalk-root: bulbous=b,club=c,cup=u,equal=e,rhizomorphs=z,rooted=r,missing=?
- stalk-surface-above-ring: fibrous=f,scaly=y,silky=k,smooth=s
- stalk-surface-below-ring: fibrous=f,scaly=y,silky=k,smooth=s
- stalk-color-above-ring: brown=n,buff=b,cinnamon=c,gray=g,orange=o,pink=p,red=e,white=w,yellow=y
- stalk-color-below-ring: brown=n,buff=b,cinnamon=c,gray=g,orange=o,pink=p,red=e,white=w,yellow=y
- veil-type: partial=p,universal=u
- veil-color: brown=n,orange=o,white=w,yellow=y
- ring-number: none=n,one=o,two=t
- ring-type: cobwebby=c,evanescent=e,flaring=f,large=l,none=n,pendant=p,sheathing=s,zone=z
- spore-print-color: black=k,brown=n,buff=b,chocolate=h,green=r,orange=o,purple=u,white=w,yellow=y

- population: abundant=a,clustered=c,numerous=n,scattered=s,several=v,solitary=y
- habitat: grasses=g,leaves=l,meadows=m,paths=p,urban=u,waste=w,woods=d

This dataset was originally donated to the UCI Machine Learning repository.and helps in answering various questions

Link https://archive.ics.uci.edu/ml/datasets.php

- What types of machine learning models perform best on this dataset?
- Which features are most indicative of a poisonous mushroom?

Python Script

1. Import the libraries
2. Import the dataset
3. Prepare the data
4. Fit the model
5. Visualize the results
6. Classify new sample

Kernel Methods

Kernel methods are a class of algorithms for pattern analysis or recognition, whose best known element is the support vector machine (SVM). The general task of pattern analysis is to find and study general types of relations (such as clusters, rankings, principal components, correlations, classifications) in general types of data (such as sequences, text documents, sets of points, vectors, images, graphs, etc)

The main characteristic of Kernel Methods, however, is their distinct approach to this problem. Kernel methods map the data into higher dimensional spaces in the hope that in this higher-dimensional space the data could become more easily separated or better structured. There are also no constraints on the form of this mapping, which could even lead to infinite-dimensional spaces. This mapping function, however, hardly needs to be computed because of a tool called the kernel trick.

1. Linear Kernel

The Linear kernel is the simplest kernel function. It is given by the inner product <x,y> plus an optional constant **c**. Kernel algorithms using a linear kernel are often equivalent to their non-kernel counterparts, i.e. KPCA with linear kernel is the same as standard PCA.

$$k(x, y) = x^T y + c$$

2. Polynomial Kernel

The Polynomial kernel is a non-stationary kernel. Polynomial kernels are well suited for problems where all the training data is normalized.

$$k(x, y) = (\alpha x^T y + c)^d$$

Adjustable parameters are the slope **alpha**, the constant term **c** and the polynomial degree **d**.

3. Gaussian Kernel

The Gaussian kernel is an example of radial basis function kernel.

$$k(x, y) = \exp\left(-\frac{||x - y||^2}{2\sigma^2}\right)$$

Alternatively, it could also be implemented using

$$k(x, y) = \exp\left(-\gamma \|x - y\|^2\right)$$

The adjustable parameter **sigma** plays a major role in the performance of the kernel, and should be carefully tuned to the problem at hand. If overestimated, the exponential will behave almost linearly and the higher-dimensional projection will start to lose its non-linear power. On the other hand, if underestimated, the function will lack regularization and the decision boundary will be highly sensitive to noise in training data.

Gaussian Radial Basis Function Kernel

It is a general-purpose kernel; used when there is no prior knowledge about the data.

Equation is:

$$k(\mathbf{x_i}, \mathbf{x_j}) = \exp(-\gamma \|\mathbf{x_i} - \mathbf{x_j}\|^2)$$

4. Sigmoid Kernel

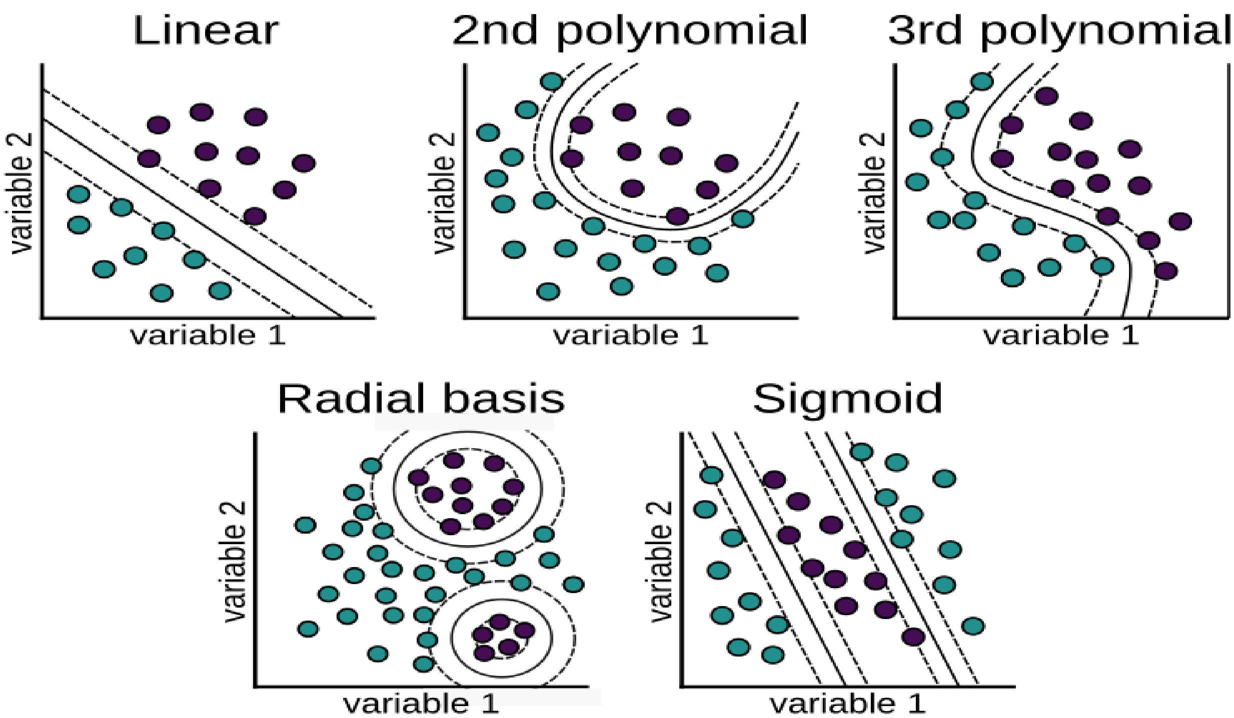
We can use it as a proxy for neural networks.

Equation is:

$$k(x, y) = \tanh(\alpha x^T y + c)$$

<u>Gamma and C parameter</u>

Cost and Gamma are the hyper-parameters that decide the performance of an SVMmodel. There should be a fine balance between Variance and Bias for any MLmode



Applications of SVM

1. Face detection – SVMc classify parts of the image as a face and non-face and create a square boundary around the face.
2. Text and hypertext categorization – SVMs allow Text and hypertext categorization for both inductive and transductive models. They use training data to classify documents into different categories. It categorizes on the basis of the score generated and then compared with the threshold value.
3. Classification of images – Use of SVMs provides better search accuracy for image classification. It provides better accuracy in comparison to the traditional query-based searching techniques.
4. Bioinformatics – It includes protein classification and cancer classification. We use SVM for identifying the classification of genes, patients on the basis of genes and other biological problems.
5. Protein fold and remote homology detection – Apply SVM algorithms for protein remote homology detection.

6. Handwriting recognition – We use SVMs to recognize handwritten characters used widely.
7. Generalized predictive control(GPC) – Use SVM based GPC to control chaotic dynamics with useful parameters

Pros and Cons associated with SVM

- **Pros:**
    - It works really well with clear margin of separation
    - It is effective in high dimensional spaces.
    - It is effective in cases where the number of dimensions is greater than the number of samples.
    - It uses a subset of training points in the decision function (called support vectors), so it is also memory efficient.
- **Cons:**
    - It doesn't perform well, when we have large data set because the required training time is higher
    - It also doesn't perform very well, when the data set has more noise i.e. target classes are overlapping
    - SVM doesn't directly provide probability estimates, these are calculated using an expensive five-fold cross-validation. It is related SVC method of Python scikit-learn library.

Conclusion: Hence in this assignment of Support Vector Machine we have implemented classification by using three types of kernel methods – Linear and Polynomial and Radial Bias Function. By default Linear Kernel is implemented without C parameter and the gamma function. Further the evaluation of accuracy obtained using polynomial and radial bias function shows that the polynomial and radial bias function performs best on no linearly separable data.