

# Major Kernel Functions in Support Vector Machine (SVM)

---

**Kernel Function** is a method used to take data as input and transform it into the required form of processing data. "Kernel" is used due to a set of mathematical functions used in Support Vector Machine providing the window to manipulate the data. So, Kernel Function generally transforms the training set of data so that a non-linear decision surface is able to transform to a linear equation in a higher number of dimension spaces. Basically, It returns the inner product between two points in a standard feature dimension.

\*\*Standard Kernel Function Equation 🤖\*

$$K(\bar{x}) = 1, \text{ if } ||\bar{x}|| \leq 1$$

$$K(\bar{x}) = 0, \text{ Otherwise}$$

**Major Kernel Functions :-**

For Implementing Kernel Functions, first of all, we have to install the "scikit-learn" library using the command prompt terminal:

```
pip install scikit-learn
```

- **Gaussian Kernel:** It is used to perform transformation when there is no prior knowledge about data.

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

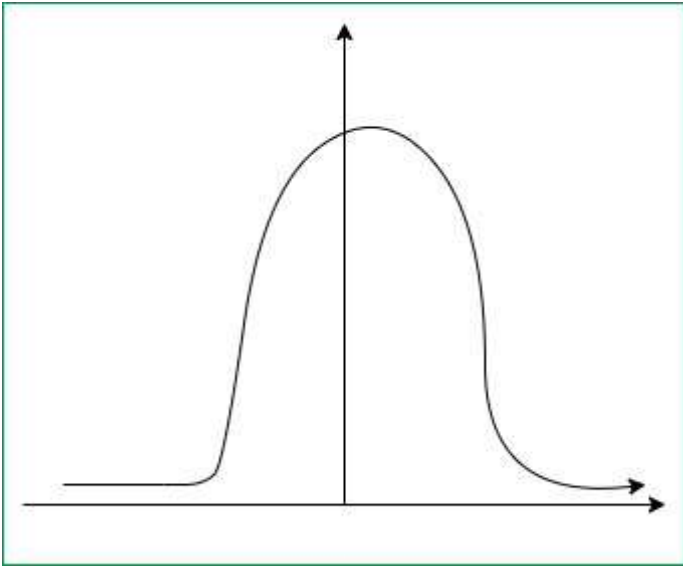
- **Gaussian Kernel Radial Basis Function (RBF):** Same as above kernel function, adding radial basis method to improve the transformation.

$$K(x, y) = e^{-\left(\gamma ||x - y||^2\right)}$$

$$K(x, x1) + K(x, x2) \text{ (Simplified - Formula)}$$

$$K(x, x1) + K(x, x2) > 0 \text{ (Green)}$$

$$K(x, x1) + K(x, x2) = 0 \text{ (Red)}$$



## Gaussian Kernel Graph

Code:

## python3

---

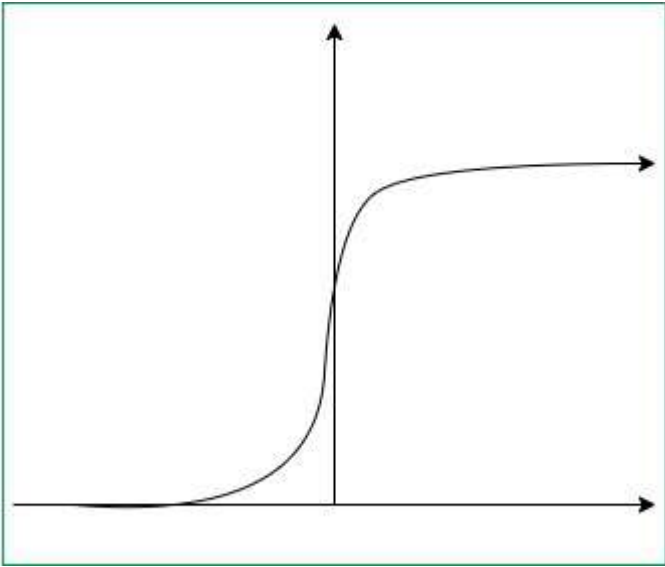
```
from sklearn.svm import SVC

classifier = SVC(kernel = 'rbf', random_state = 0)

classifier.fit(x_train, y_train)
```

- **Sigmoid Kernel:** this function is equivalent to a two-layer, perceptron model of the neural network, which is used as an activation function for artificial neurons.

$$K(x, y) = \tanh(\gamma \cdot x^T y + r)$$



Sigmoid Kernel Graph

Code:

## python3

---

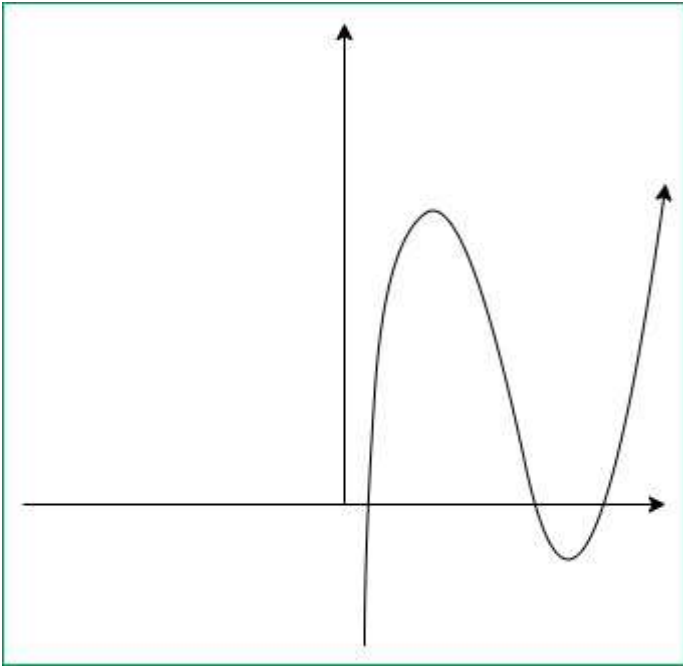
```
from sklearn.svm import SVC

classifier = SVC(kernel = 'sigmoid')

classifier.fit(x_train, y_train)
```

- **Polynomial Kernel:** It represents the similarity of vectors in the training set of data in a feature space over polynomials of the original variables used in the kernel.

$$K(x, y) = \tanh(\gamma \cdot x^T y + r)^d, \gamma > 0$$



## Polynomial Kernel Graph

Code:

python3

---

```
from sklearn.svm import SVC

classifier = SVC(kernel = 'poly', degree = 4)

classifier.fit(x_train, y_train)
```

- **Linear Kernel:** used when data is linearly separable.

Code:

python3

---

```
from sklearn.svm import SVC

classifier = SVC(kernel = 'linear')

classifier.fit(x_train, y_train)
```