

**Amrita Viswa Vidyapeetham**

Centre for Excellence in Computational Engineering Networking Coimbatore - 641-112 (India)

**Kernel Methods and its Applications**

**Prepared By:**

**Group-11**

P.S.S.Sai Keerthana (CB.EN.U4AIE21038)

P.Komal Sai Anurag(CB.EN.U4AIE21039)

Udaygiri Varun (CB.EN.U4AIE21071)

Sejal Singh (CB.EN.U4AIE21061)

**Supervised By:**

**Dr.Neethu Mohan**

Asst. Professor

A 3rd Semester Project submitted to the Department of Centre for Excellence in Computational and Networking partial fulfillment of the requirement for final project of Python for Machine learning

January, 2023

**Amrita School of Engineering**

**Amrita Viswa Vidyapeetham**

Coimbatore - 641-112 (India)

**Declaration**

We declare that the submitted report is our original work and no values and context of it has been copy pasted from anywhere else. We take full responsibility, that if in future, the report is found invalid or copied, the final decision will be of concerned faculty. Any form of plagiarism will lead to disqualification of the report.

Place: Ettimadai

Date: 26.01.2023

**Acknowledgement**

We are extremely grateful to our professor Dr.Neethu Mohan, for this opportunity to work on this project. Her exceptional teaching skills and guidance were crucial in our ability to implement kernel methods and its applications. Her knowledge and support were essential in leading us through the project and fostering our enthusiasm to see it to successful completion. We are deeply appreciative of the contribution of Prof. B. Ganga Gowri to our project.

**Abstract**

Kernel methods are a type of machine learning algorithm that map data into a higher dimensional feature space. These methods have found widespread applications in areas such as pattern recognition, computer vision, and natural language processing. Despite their popularity, there are some limitations of kernel methods, including high computational cost and the need for careful selection of the kernel function. Despite these limitations, kernel methods continue to be an important tool for researchers and practitioners alike, with new developments and improvements constantly being made

**INTRODUCTION**

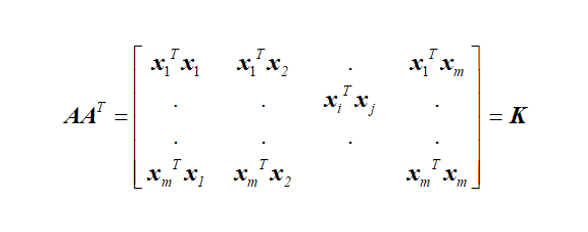
**What are Kernels?**

Algorithms used for pattern analysis are called kernel methods. These techniques rely on linear classifiers to address nonlinear issues. Kernel methods are essentially algorithms that enable implicit data projection in a high-dimensional domain.

This is achieved by using a kernel function, which is a mathematical function that takes as input two data points and returns a scalar value that represents the similarity between them.

The mathematical expression of a kernel function is typically a dot product between the inputs in a transformed feature space, which is often called the "kernel space" and then find the optimal hyperplane that separates the different classes. This hyperplane is chosen in such a way as to maximize the margin, which is the distance between the hyperplane and the closest data points from either class. The data points that are closest to the hyperplane are called support vectors and play a crucial role in determining the position of the hyperplane.

The dot product in the original input space is transformed into a dot product in the kernel space by the kernel function.



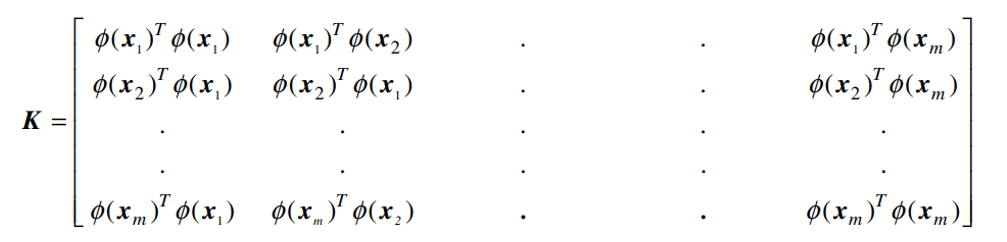
Note that (i,j) element of AAT is xiT,xj. The matrix is called linear Kernel Matrix.

Support Vector Machine (SVM), also known as support vector networks, is a category of machine learning technique for classification and regression in supervised learning. Instead of regression, classification problems are where SVM is most frequently used.

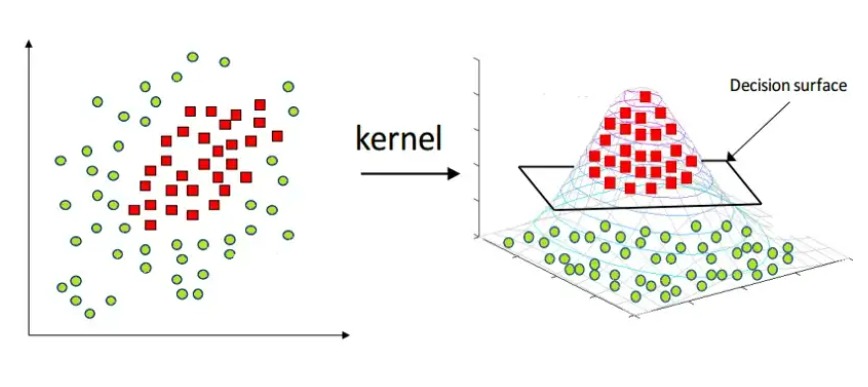
The term "Kernel" is used because the Support Vector Machine uses a collection of mathematical operations to provide the window through which the data can be manipulated.

In order for a non-linear decision surface to turn into a linear equation in a higher number of dimension spaces, Kernel Function often changes the training set of data. The inner product between two points in a common feature dimension is what it basically returns.

φ (.)🡪 x A non-linear mapping function that maps input vector x into a high dimensional feature vector



Each data point x; can be mapped into a higher-dimensional space using the function φ(.), and the maximum separating hyperplane in that space can then be used as a classifier. The SVM algorithm for training needs φ(xi)T, φ(xj) for all i and j if such a separation is possible. The processing need is enormous if such explicit mapping is carried out and the dot product is mapped in the higher dimensional space, particularly when either the data dimension, the number of data points, orboth are significant.



**Types of kernel methods:**

Support vector machines use various kinds of kernel methods. Here are a few of them which are significant:

* Linear Kernels
* Polynomial Kernels
* RBF Kernels
* Laplace Kernels
* Sigmoid Kernels

Let us look into the brief elaboration of the above mentioned kernel methods.

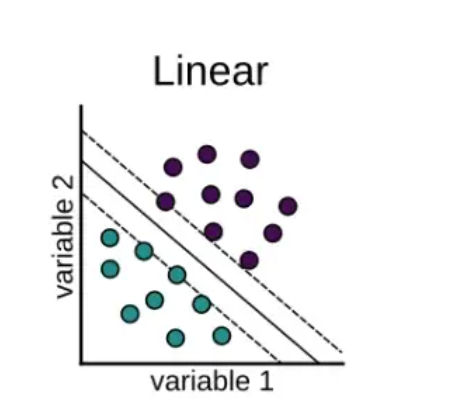
**Linear Kernels:**

When the data can be split using a single line, or when it is linearly separable, a linear kernel is utilised. It is one of the most often adopted kernels. It is typically employed when a given data set has a sizable number of features. Text Classification is one of the instances where there are numerous features because each letter of the alphabet is a separate feature. Therefore, Linear Kernel is primarily used in Text Classification.

The linear kernel is good when there is a lot of features.

If there are two kernels named x1 and x2, the linear kernel can be defined by the dot product of the two vectors:

K(x1, x2) = x1 . x2

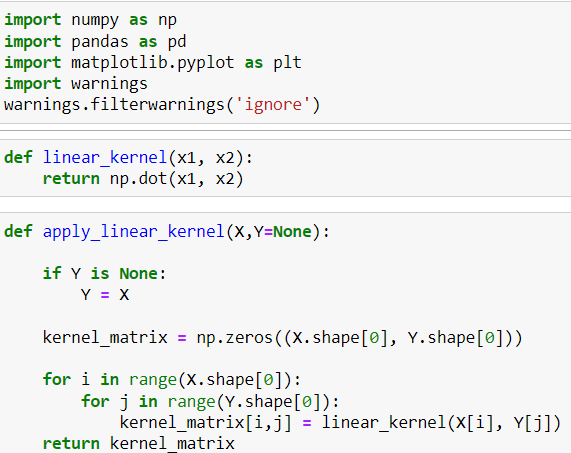


**Advantages:**

Linear kernel is faster

Less parameters to optimize

**IMPLEMENTATION:**

****

**Polynomial Kernel methods:**

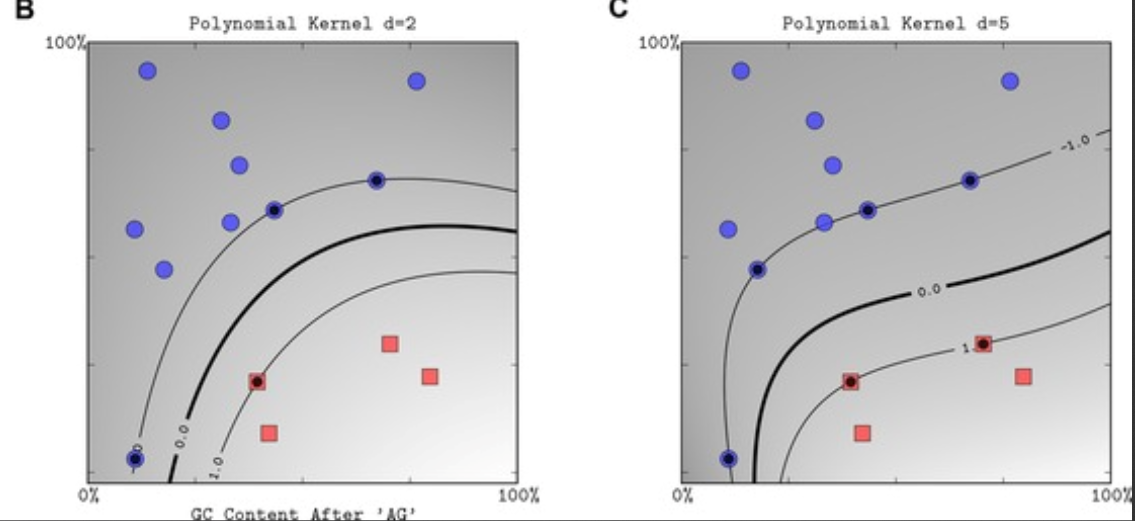
The polynomial kernel, a kernel function in machine learning that is frequently used with support vector machines (SVMs) and other kernelized models, represents the similarity of vectors (training samples) in a feature space over polynomials of the original variables, enabling the learning of non-linear models.

It is defined as the dot product of the input vectors raised to a power, typically represented as

K(x1, x2) = (x1 . x2 + c)d

where x1 and x2 are input vectors, c is a constant and d is the degree of the polynomial. Vectors of features computed from training or test samples and c ≥ 0 is a free parameter trading off the influence of higher-order versus lower-order terms in the polynomial.

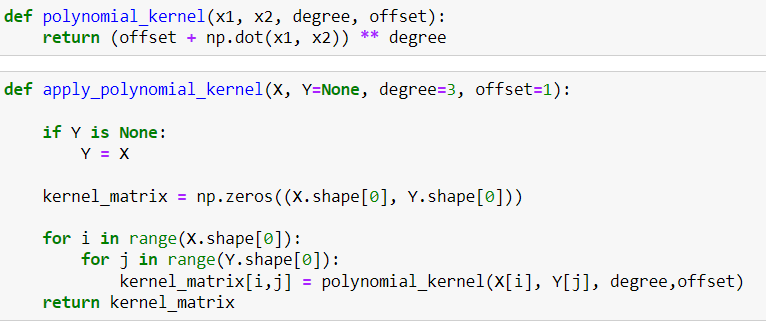
When c = 0, the kernel is called homogeneous.



**Advantages:**

Features derived from training or test samples, and c 0 is a free parameter that balances the impact of the polynomial's higher-order and lower-order terms. The kernel is said to as homogenous when c = 0. (An additional generalised polykernel divides xTy by a scalar parameter that is provided by the user.

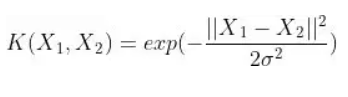
**IMPLEMENTATION:**



**RBF Kernels:**

A RBF (Radial Basis Function) kernel is employed in place of the conventional linear kernel, which presumes that the boundaries between the various classes are linear and are defined by a straight line.

The similarity or degree of proximity between two points, X1 and X2, is calculated using the RBF kernel function. This kernel has the following mathematical representation:



where,

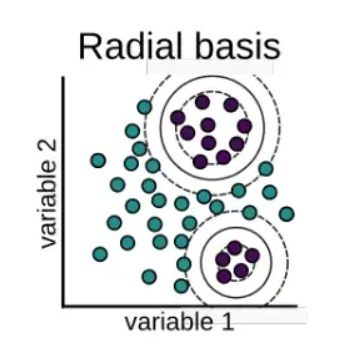
1. ‘σ’ is the variance and our hyperparameter

2. ||X₁ - X₂|| is the Euclidean (L₂-norm) Distance between two points X₁ and X₂

The RBF kernel can only take on a maximum value of 1, which happens when two data points, X1 and X2, are equal.

When the points are identical, there is no space between them, making them incredibly similar.

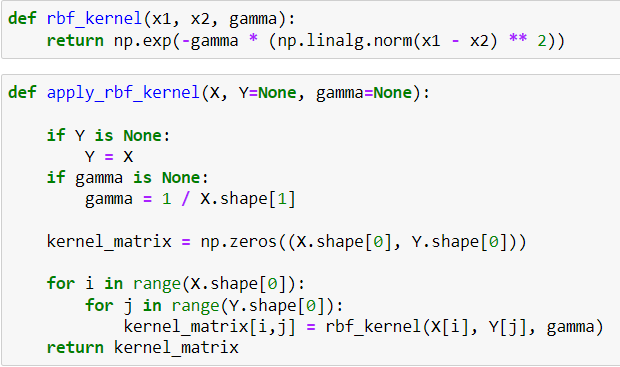
The kernel value is less than 1 and near to 0 when the distance between the points is great, indicating that the points are dissimilar.



**Advantages:**

From a huge number of data points, smooth surfaces are generated using RBFs. When applied to surfaces with gentle variations, like height, the functions provide good results.

**IMPLEMENTATION:**

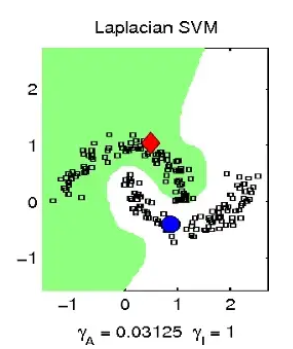


**Laplace Kernels:**

A kernel function utilised in many machine learning techniques, including support vector machines (SVMs) and Gaussian processes, is the Laplace kernel, also referred to as the exponential kernel. It is expressed as

K(X,Y)=exp(-||x-y||/σ)

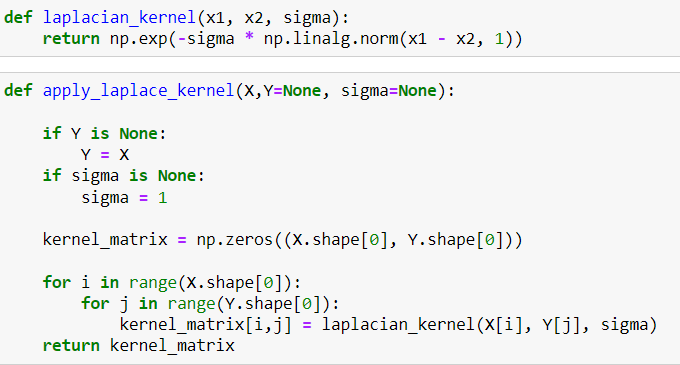
where x and y are input vectors and is the kernel width or scale parameter. It is defined as the exponential of the negative absolute difference of the input vectors.



Advantages:

A Laplacian kernel is less prone to changes.

**IMPLEMENTATION:**



**Sigmoid Kernel method:**

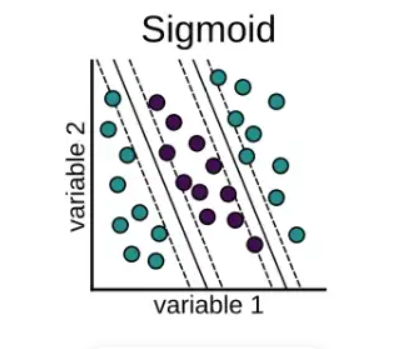
The sigmoid kernel, also referred to as the hyperbolic tangent kernel, is a kernel function that is utilised in neural networks and support vector machines (SVMs), among other machine learning algorithms. It can be written as

K(x,y) = tanh(k(xy)+c)

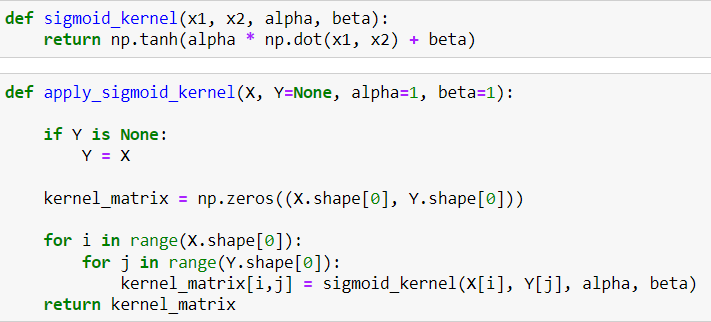
where x and y are input vectors, k is a scalar, and c is a bias parameter. It is defined as the hyperbolic tangent of the dot product of the input vectors multiplied by a scalar.

**Advantages:**

When employed as an activation function for synthetic neurons, this function is comparable to a two-layer perceptron neural network model.



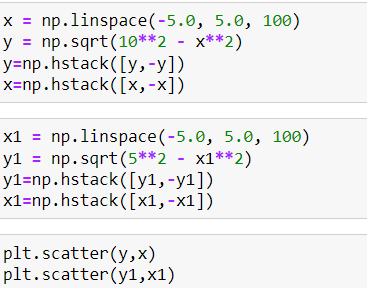
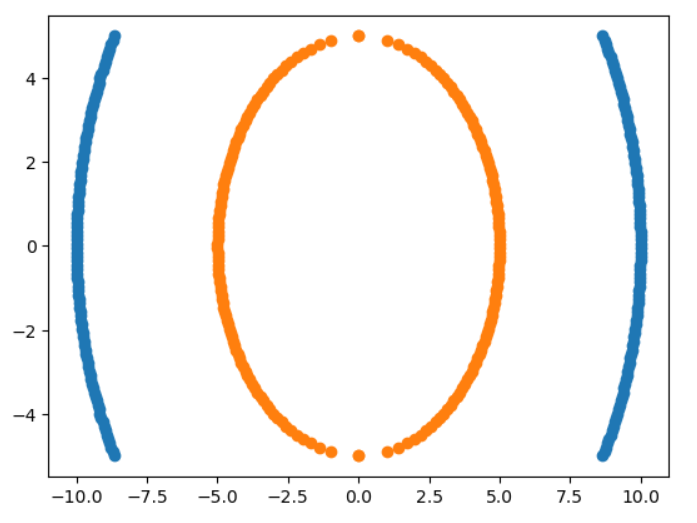
**IMPLEMENTATION:**

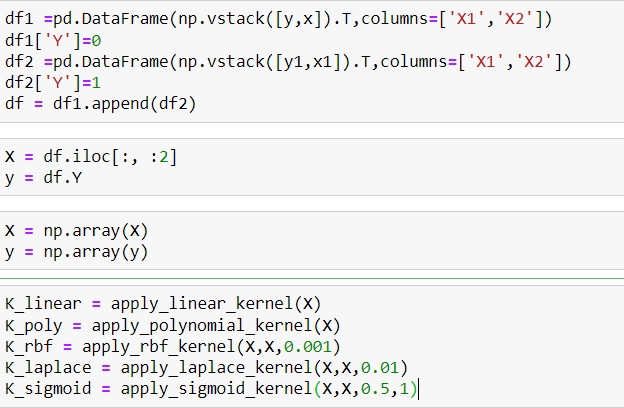


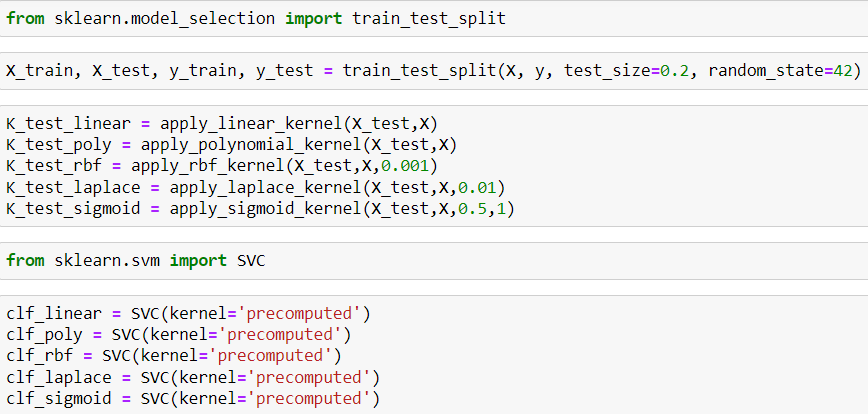
**Implemented Applications using Kernels:**

**Classification in SVM:**

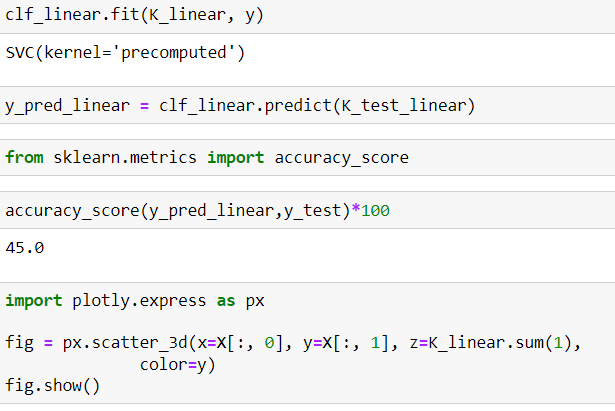
We have created a non-linear data and separated the data using all the kernels that we implemented and displayed the 3D-visualization of every kernel

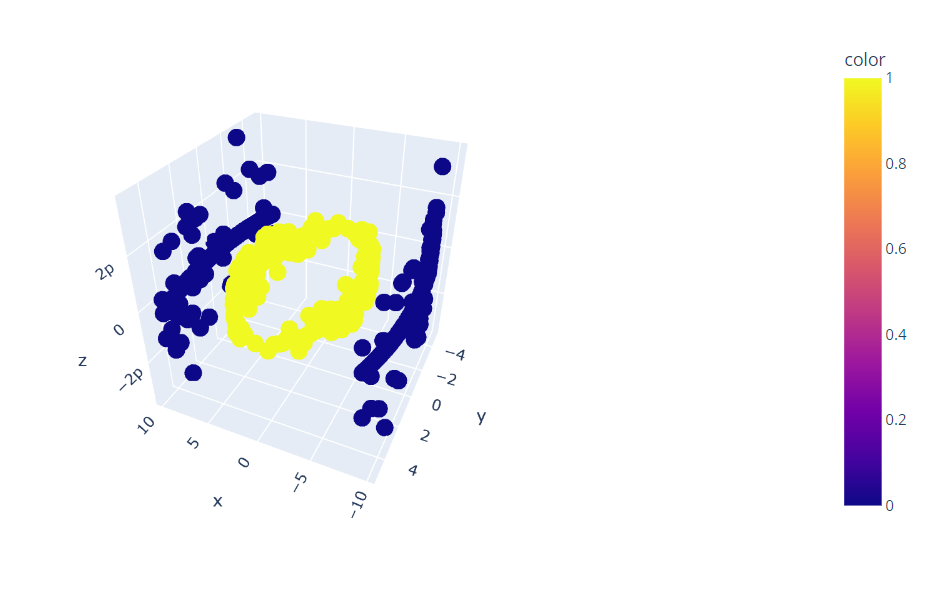




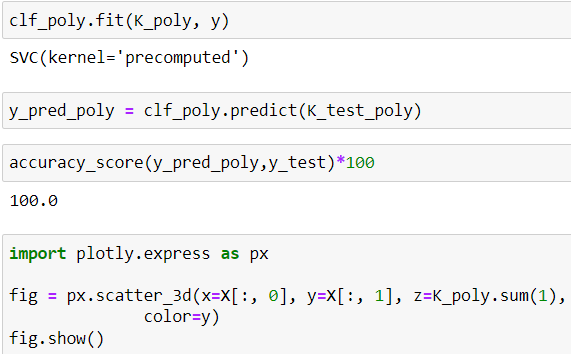
**Classification using Linear Kernel:**



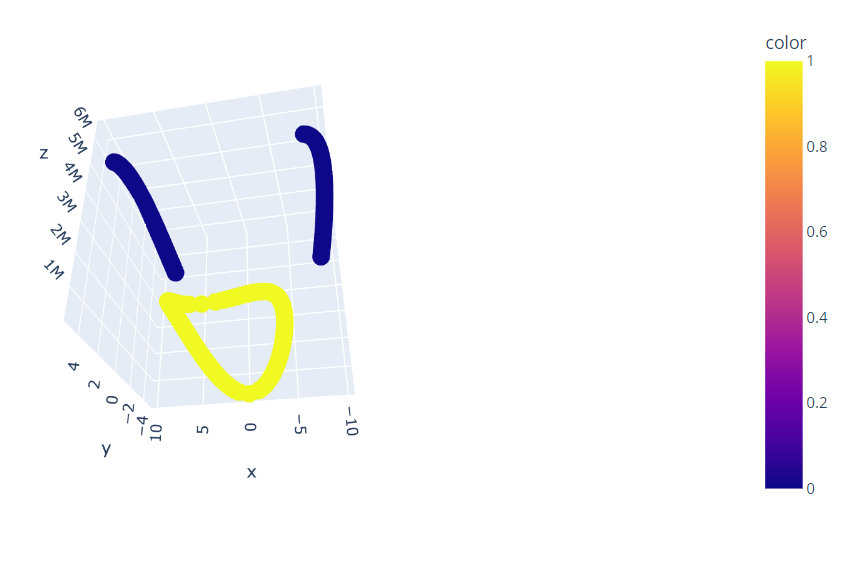
**3D Visualisation:**



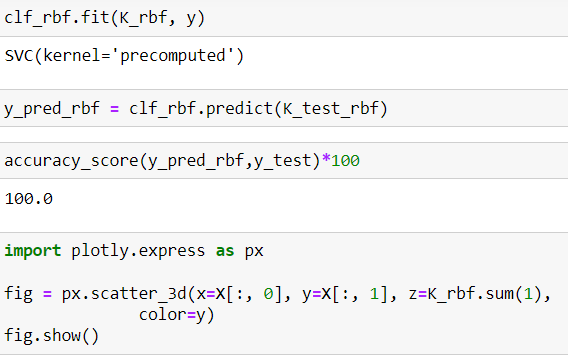
**Classification using Polynomial Kernel:**



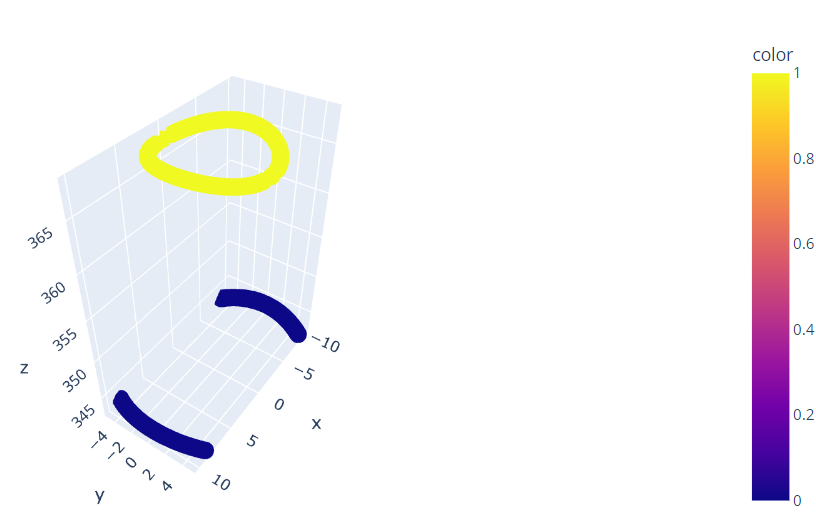
**3D Visualisation:**



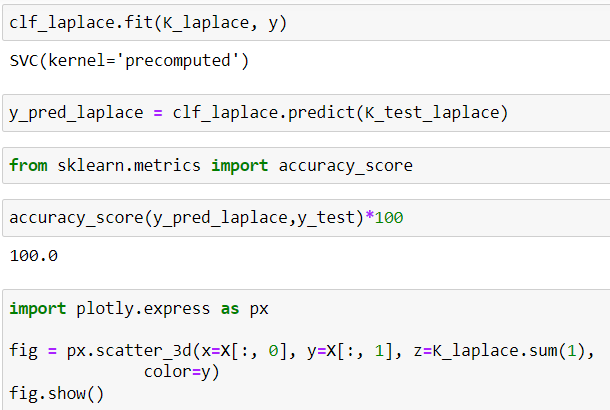
**Classification using RBF kernels:**



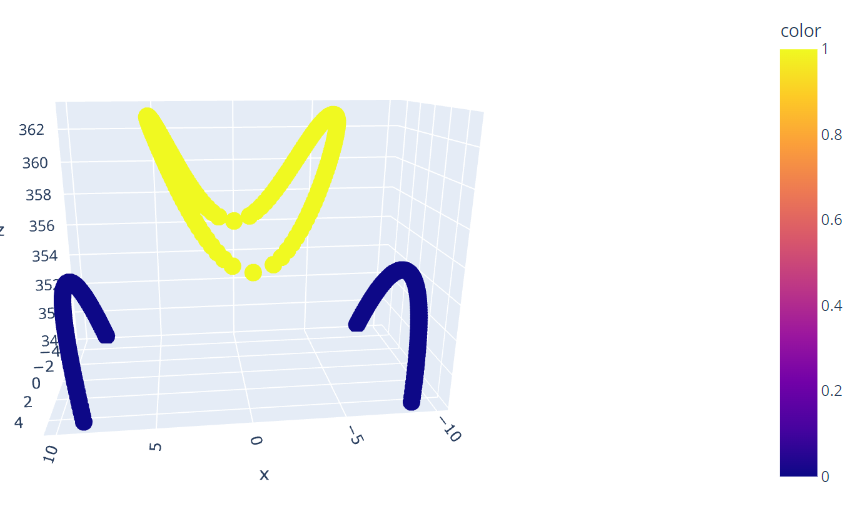
**3D Visualisation:**



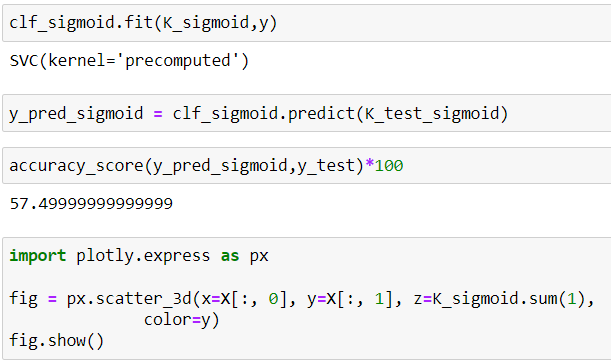
**Classification using Laplace kernel:**



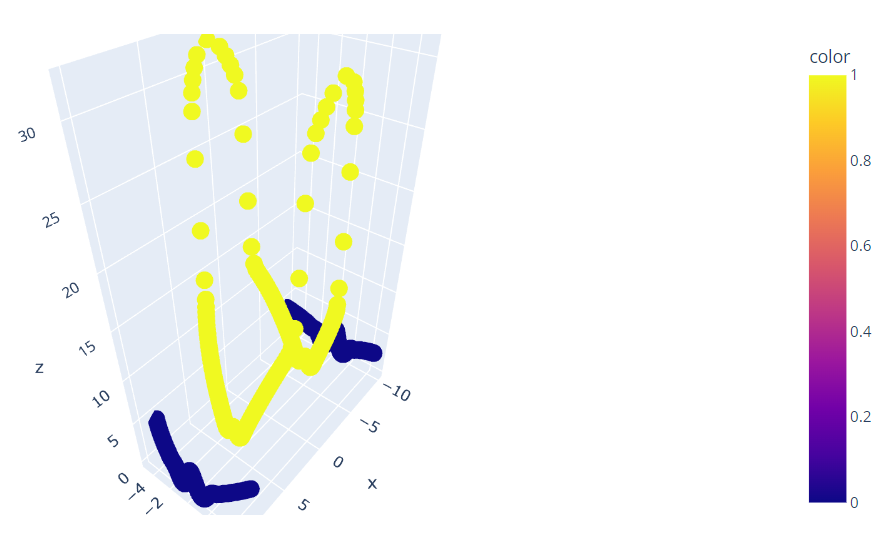
**3D Visualisation:**



**Classification using Sigmoid Kernel:**



**3D Visualisation:**



**IMPLEMENTATION OF KERNEL PCA:**

Data classification can be accomplished using the statistical method known as Principal Component Analysis (PCA). PCA can be used to simplify classification by reducing the dimensionality of the data. The data can be projected onto the first few primary components, which often capture the most crucial aspects of the data, to achieve this.

*Cons of PCA for classification:*

PCA is a linear method, so it's vital to remember that non-linear data may not respond well to it. Additionally, PCA is sensitive to the data's scaling, thus it's critical to check that the data has been scaled correctly before using PCA.

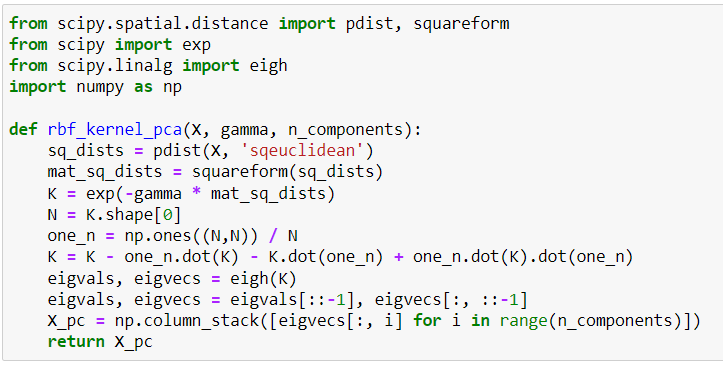
**Kernel PCA:**

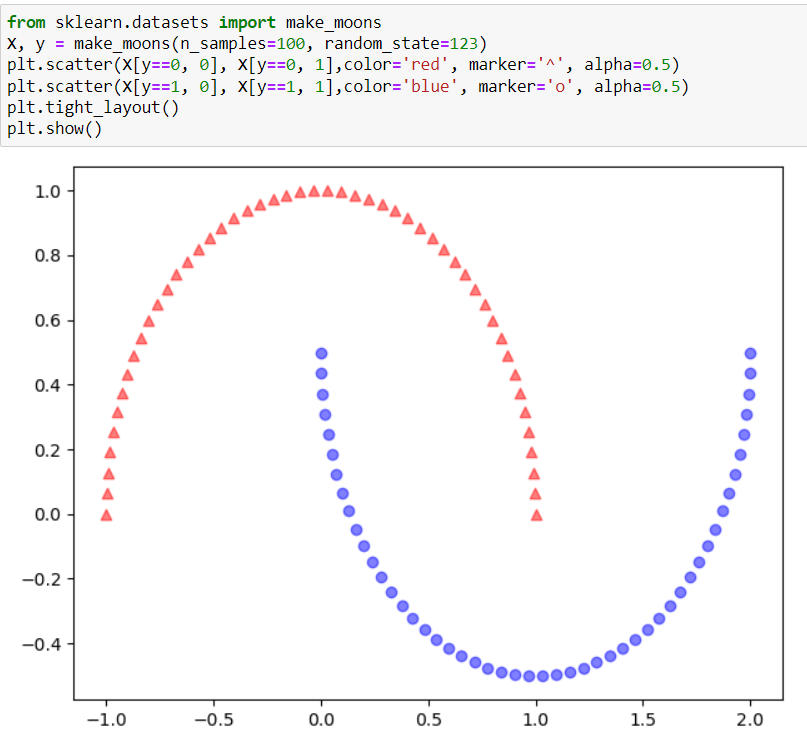
For classification problems, kernel principal component analysis (Kernel PCA) can be employed. The main concept is to first reduce the dimensionality of the data using Kernel PCA, and then classify the data using any classification algorithm, such as support vector machines (SVMs), k-nearest neighbour (k-NN), decision trees, etc., based on the principal components identified by Kernel PCA.

*Advantanges of kernel PCA*:

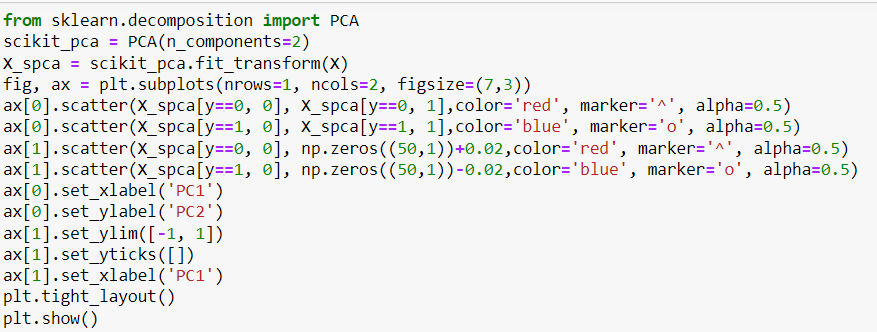
The key benefit of utilising Kernel PCA for classification is that it can enhance classifier performance by lowering the dimensionality of the data while maintaining the non-linear correlations between the features. By doing this, the classifier may become more reliable and less prone to overfitting.

**IMPLEMENTATION:**

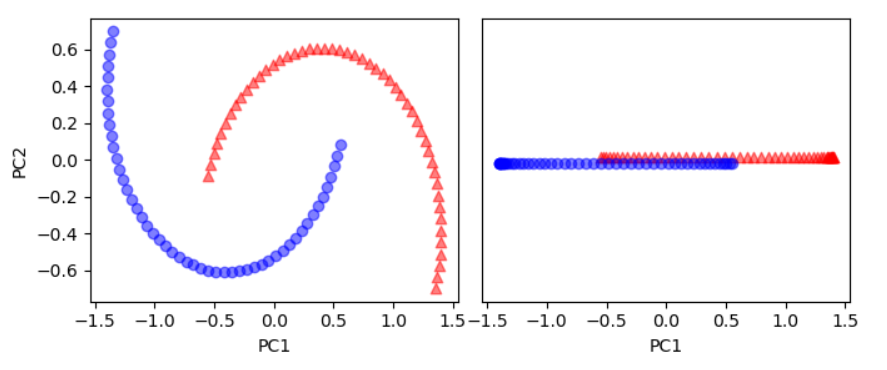


\

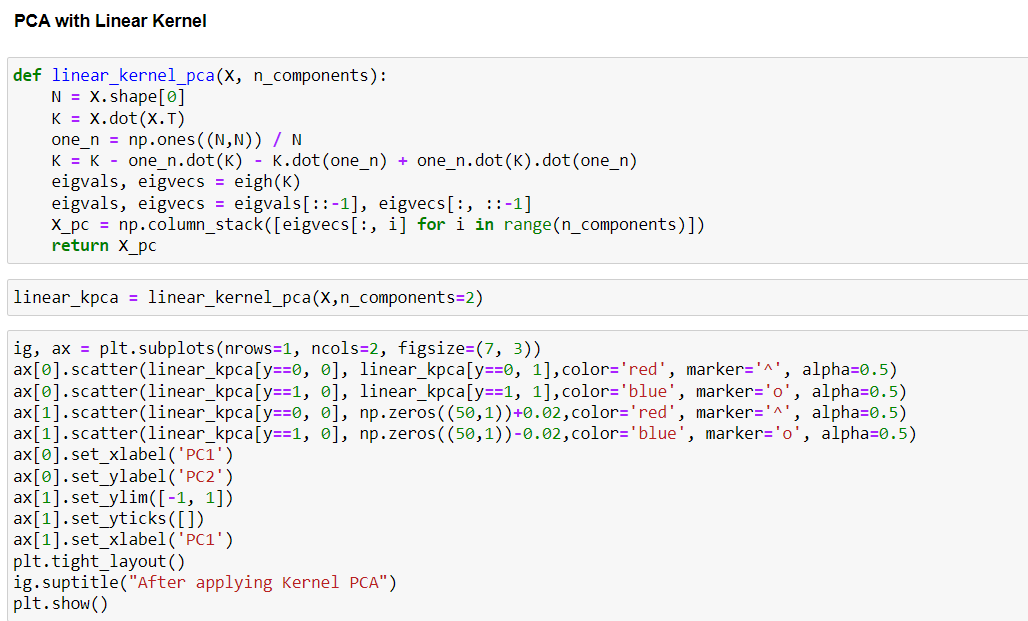
**CLASSIFICATION USING PCA:**



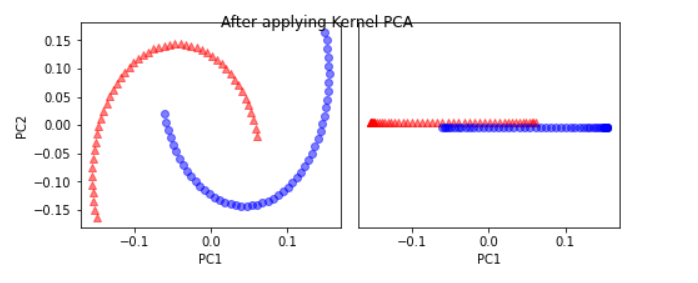
**VISUALISATION OF SEPERATION USING PCA:**



**Classification uisng Kernel PCA( Linear Kernel):**

****

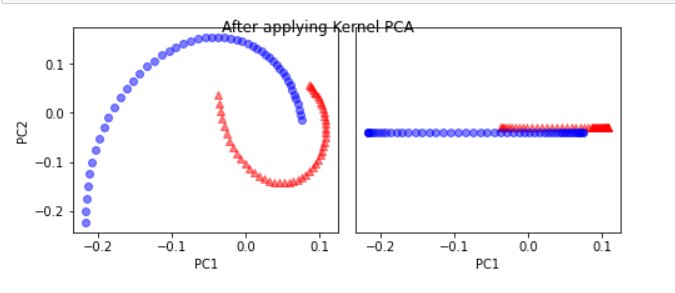
**VISUALISATION OF SEPERATION USING PCA:**

****

**Classification uisng Kernel PCA( Polynomial Kernel):**

****

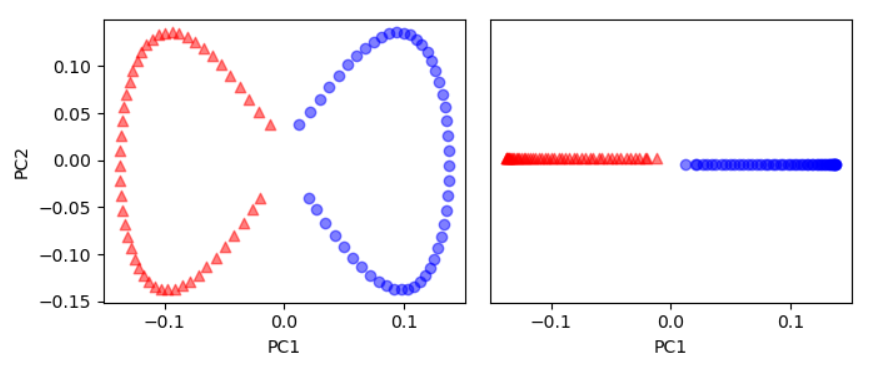
**VISUALISATION OF SEPERATION USING PCA:**

****

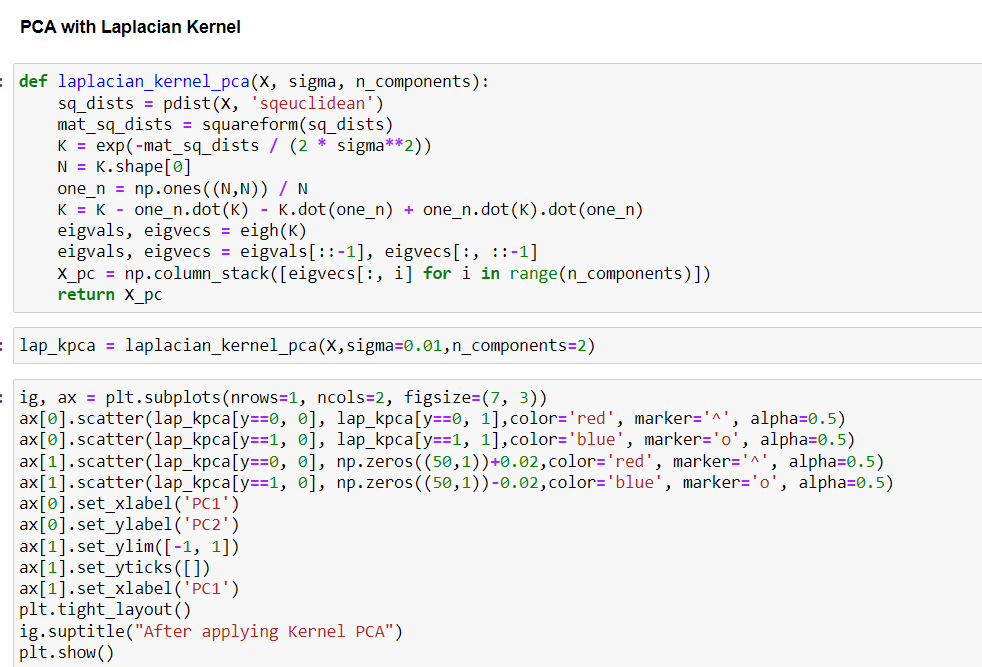
**Classification uisng Kernel PCA( RBF Kernel):**



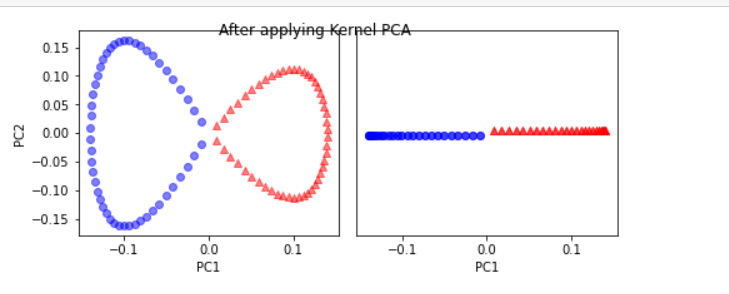
**VISUALISATION OF SEPERATION USING PCA:**



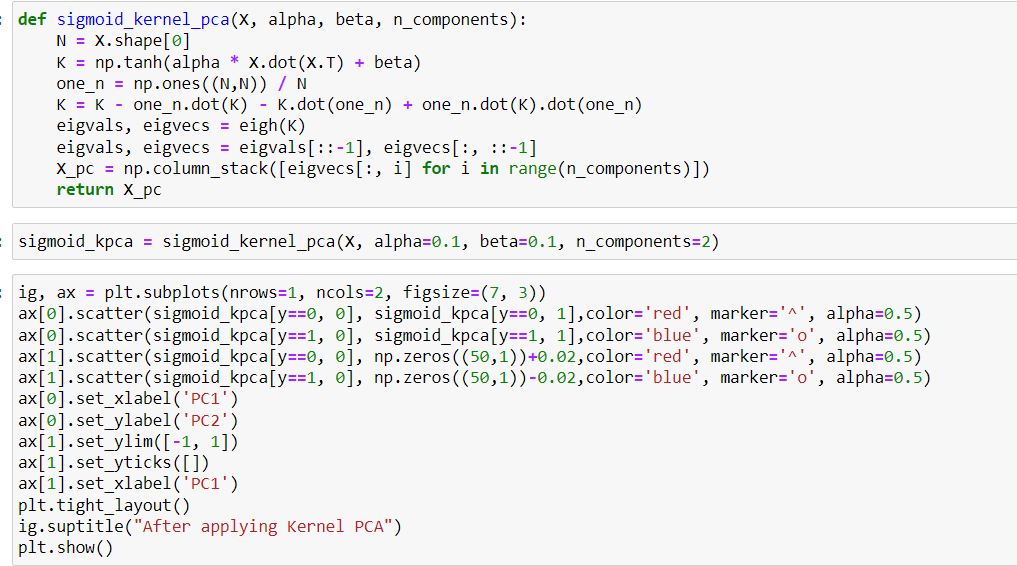
**Classification uisng Kernel PCA( Laplacian Kernel):**

****

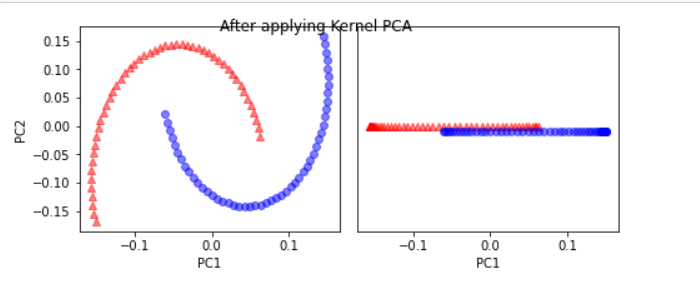
**VISUALISATION OF SEPERATION USING PCA:**

****

**Classification uisng Kernel PCA(Sigmoid Kernel):**

****

**VISUALISATION OF SEPERATION USING PCA:**

****

**Challenges using Kernel PCA:**

The choice of kernel function has an impact on Kernel PCA, so it's critical to select one that is appropriate for the particular dataset and classification task. It's vital to select the right number of principal components that strikes a balance between dimensionality reduction and the preservation of information because the number of principal components utilised may also have an impact on the performance of the classifier.

**Applications of Kernel Methods:**

1.Gaussian Processes:

A probabilistic model that can be applied to classification and regression problems. They can represent sophisticated, non-linear interactions because they use kernel functions to specify the covariance structure of the data.

2. Image processing:

Image processing operations like edge identification, feature detection, and image segmentation require kernel algorithms. Kernels are used by edge detection algorithms to identify borders and edges in images. Kernels are used by feature detection algorithms to find patterns and features in images. Kernels are used by image segmentation algorithms to separate a picture into separate sections. Used in CNN kernel-based operations to learn the features of the data.

3. Image Classification:

Using kernel methods, such as support vector machines (SVMs), to categorise images according to their attributes is known as "image classification with kernels." Without having to compute the coordinates of the images in that space, the kernel approach is utilised to calculate the dot product of the images in the higher-dimensional space. This increases the computational efficiency of image classification using kernels.

4. Bioinformatics:

Bioinformatics applications like protein structure prediction and gene expression analysis can also make use of kernel approaches.

The process of figuring out the three-dimensional structure of a protein from its amino acid sequence is one example of employing kernel approaches for protein structure prediction. By comparing a protein's sequence to a list of known protein structures, kernel algorithms can be used to predict a protein's structure. More precise predictions can be made by using the kernel method to determine how similar the sequences are in a high-dimensional space.

Another illustration is the application of kernel approaches to gene expression analysis, which is the investigation of gene activity in various biological materials. Genes that differ in their levels of activity or expression between samples can be found using kernel approaches. The kernel method can be used to determine how comparable the gene expression profiles are in a high-dimensional space, making it possible to identify differentially expressed genes with more accuracy.

Additionally, bioinformatics applications like drug discovery and the prediction of protein-protein interactions can make use of kernel approaches.

5. Robotics:

Control, perception, and planning are just a few robotics applications where kernel approaches can be applied.

One instance is the creation of nonlinear controllers for robotic systems employing kernel approaches for control. Kernel-based controllers have the capacity to manage complicated phenomena and deliver reliable performance even in the presence of model uncertainty and disturbances.

Another illustration is the usage of kernel approaches for perception, which can be applied to robot localisation, scene interpretation, and object recognition. In order to classify and cluster the data for object recognition and scene interpretation, kernel algorithms can be employed to extract features from photos and sensor data. In addition, they can be used to estimate the robot's pose for localization and to match sensor data with a map.

Overall, complex and nonlinear computations in robotics can be carried out using kernel approaches, enabling more reliable and precise performance in a variety of applications.