Classification of Data

PR_2020 Project_01

Group Details:

A.Hemanth Rao	M.Pavan Karthik	P.V.S.VamseeKrishna
(S2018020200)	(S20180020218)	(S20180020228) .

Description : A detailed report of classification of data using Non-linear Transformation and Direct Non-linear classification.

1. Introduction:

In this project we are going to do non-linear transformation and direct non-linear classification with non-linear transformation divider data set and sine lock data set. In non-linear classification we use models like linear discriminant, quadratic discriminant, Support vector machine with gaussian kernel, support vector machine with polynomial kernel and K-nearest neighbour classifier will be done using the Classifier app and study their analysis with their accuracy.

2. Problem Description:

2.1 Classification of two data sets named as Class1 and Class2.

- Class 1 : $x \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$

  $\boldsymbol{\mu} = [\,\boldsymbol{\mu 1}\,, \boldsymbol{\mu 2}\,]^T$ $\boldsymbol{\Sigma}$ = [15 0 ; 0 1]

- Class 2 : $x \sim \mathcal{N}(m, L)$

  $\mathbf{m} = [\,\boldsymbol{\mu 3}\,,\ \pm\boldsymbol{\mu 4}\,]^T$ $\mathbf{L} = \sigma^2 I_{2x2}$

2.2 Classification of two data sets named as Class1 and Class2.

- Class 1 : $x \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$

  $\boldsymbol{\mu} = [\ h_1\ +\ acos(t),\ bsin(t)\ ]^T$
  $\boldsymbol{\Sigma} = \sigma^2 I_{2x2}$

- Class 2 : $x \sim \mathcal{N}(m, L)$

  $\mathbf{m} = [\ h_2 + acos(t), k - bsin(t)\ ]^T$
  $\mathbf{L} = \boldsymbol{\Sigma}$

3. Methodology:

3.1 Problem 1:

- Scaling down Feature_1 of both the classes by using a Scaling factor.
- Using Euclidean Distance concept*[1]* ,Converting of 2D- data to 1D- data..
- Using Least squares method *[2]* for building a Model.
- Evaluating the model performance using Confusion Matrix*[3]*.
- Reconstruction of Threshold in 2D by inverting equation.

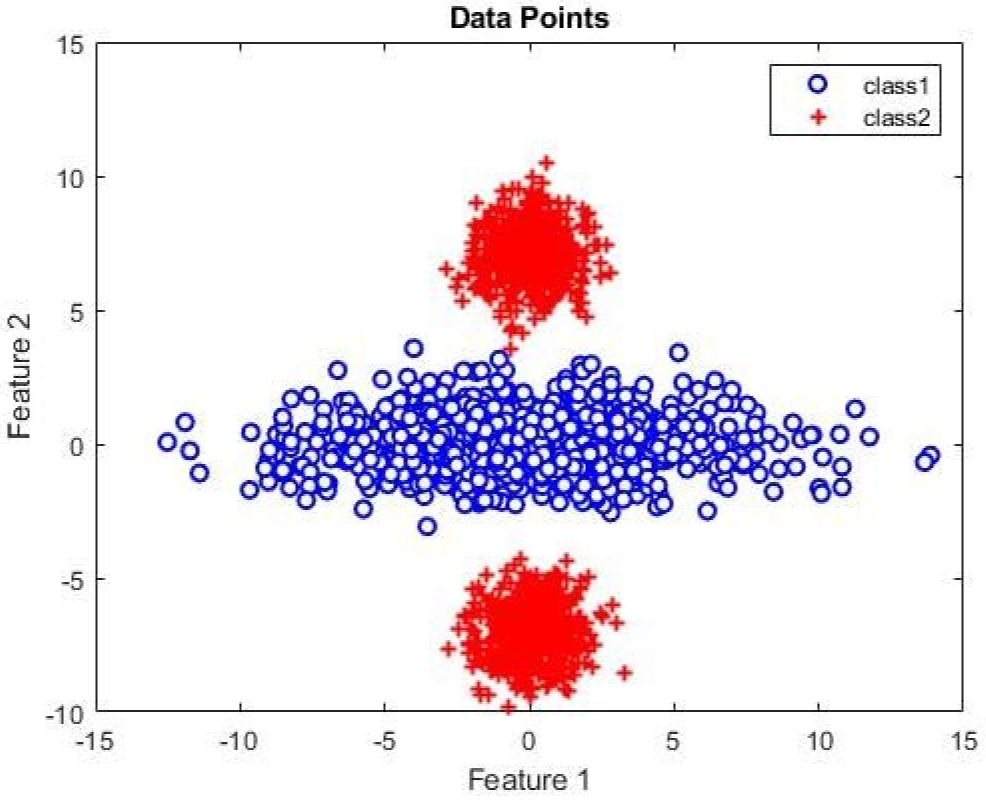
3.2 Problem 2:

- Using in built applications for classification of the given with different classifiers.
- Analyzing and comparing the output of each by using Accuracy metric.

4. Implementation and Results:

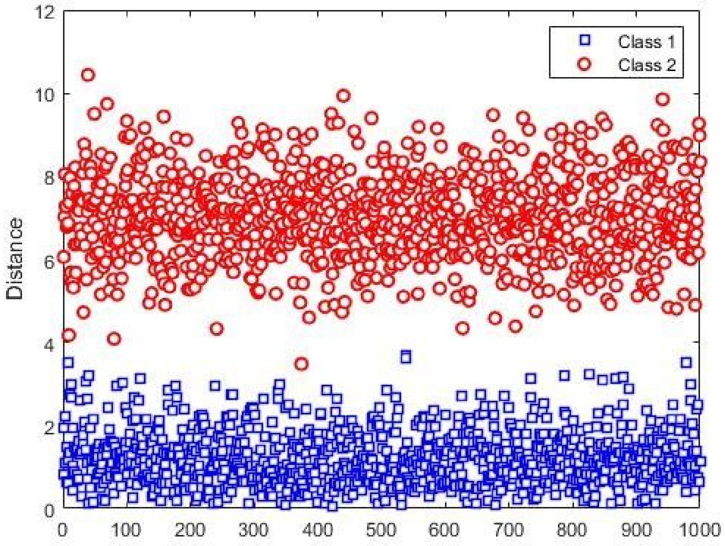
4.1 Problem 1:

- Input Data.

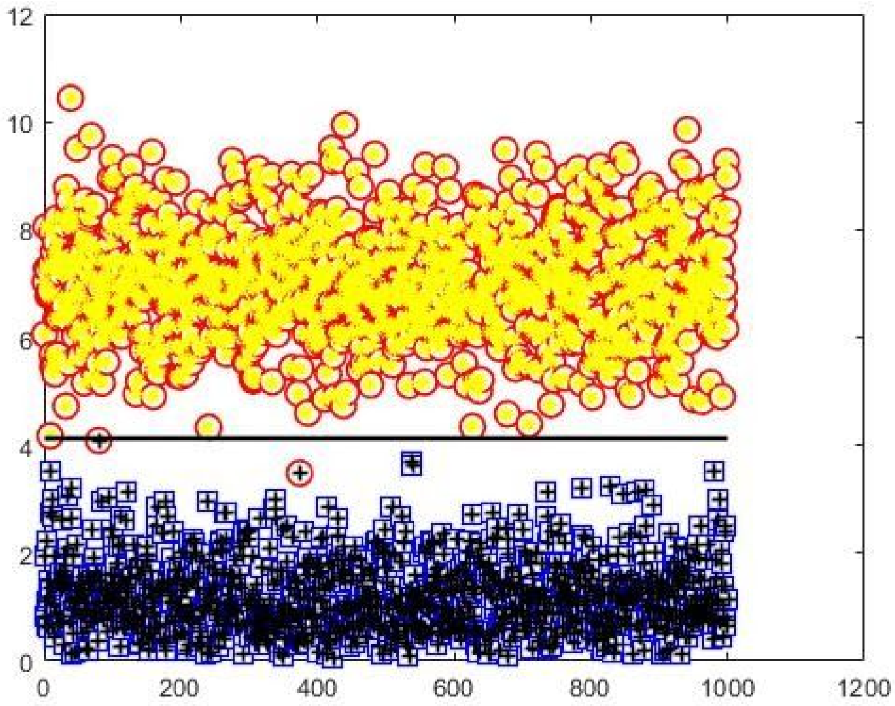


- Scaling down the Feature_1 by a scaling factor M , Conversion of 2D data into 1D-Data using Euclidean Distance.

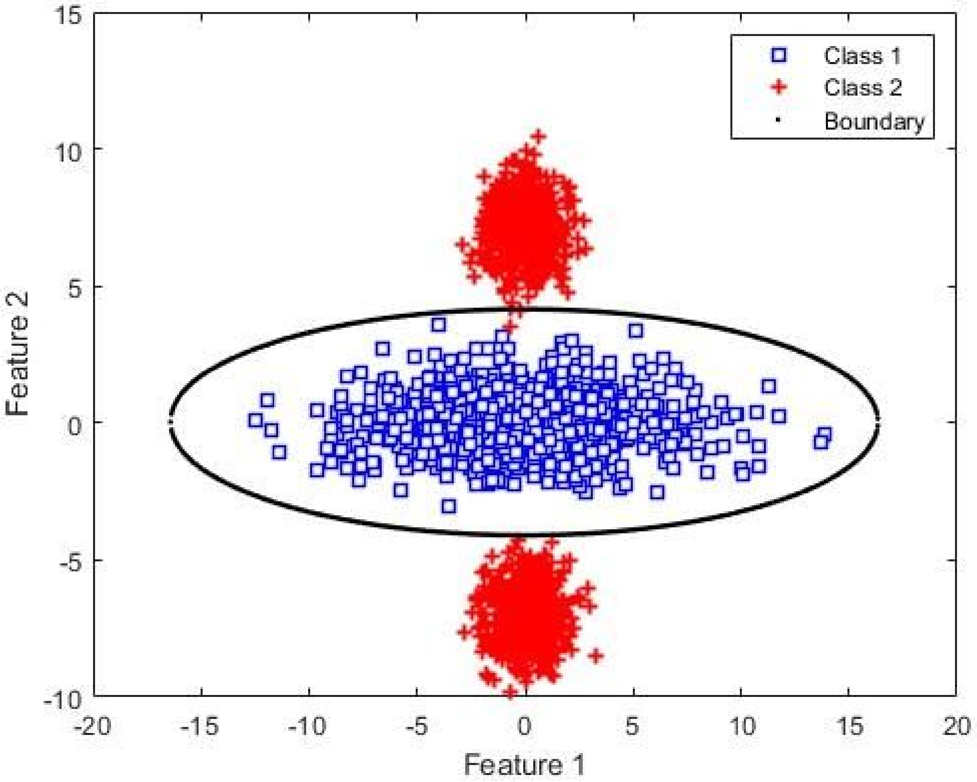
Where, $$M = \frac{max\{class_1(f_2)\} - min\{class_2(f_2)\}}{max\{class_1(f_1)\} - min\{class_2(f_1)\}}$$



- Building Model by using Least Squares method, Evaluating the Model using Confusion matrix.

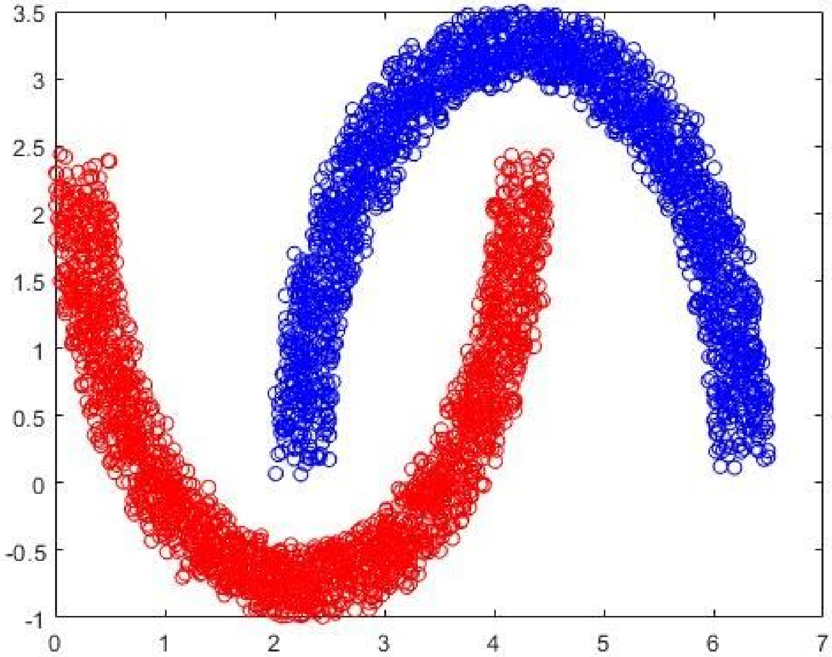


- Construction of Decision Boundary in 2D space and Scaling Up the Boundary by M.

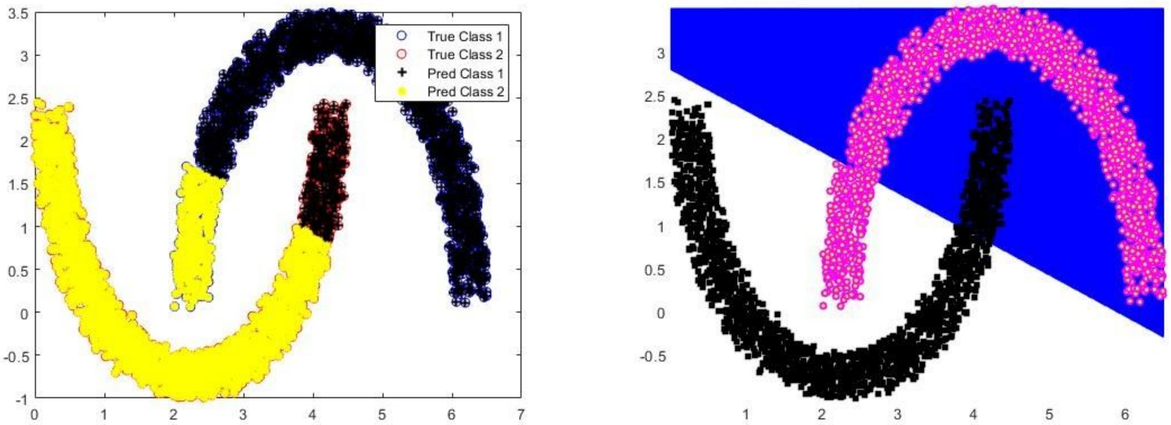


4.2 Problem 2:

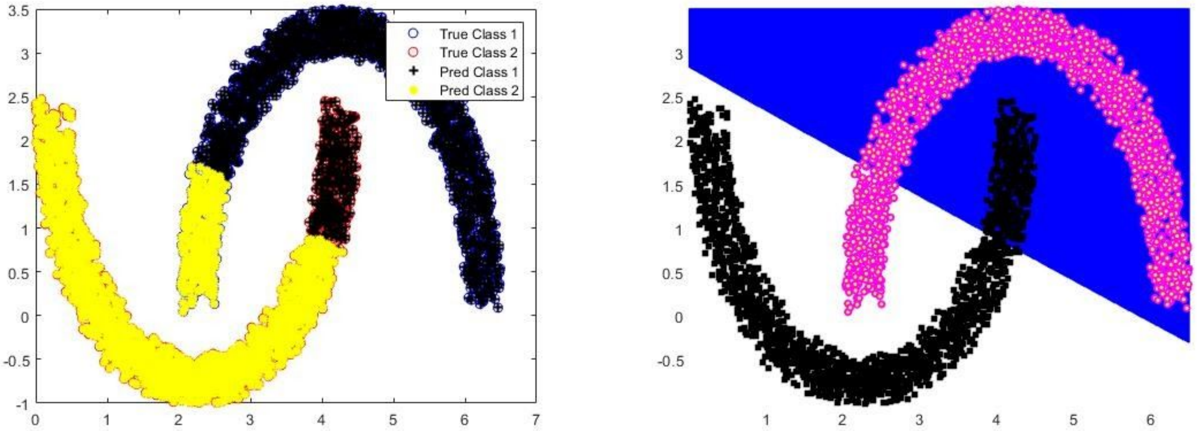
- Input data



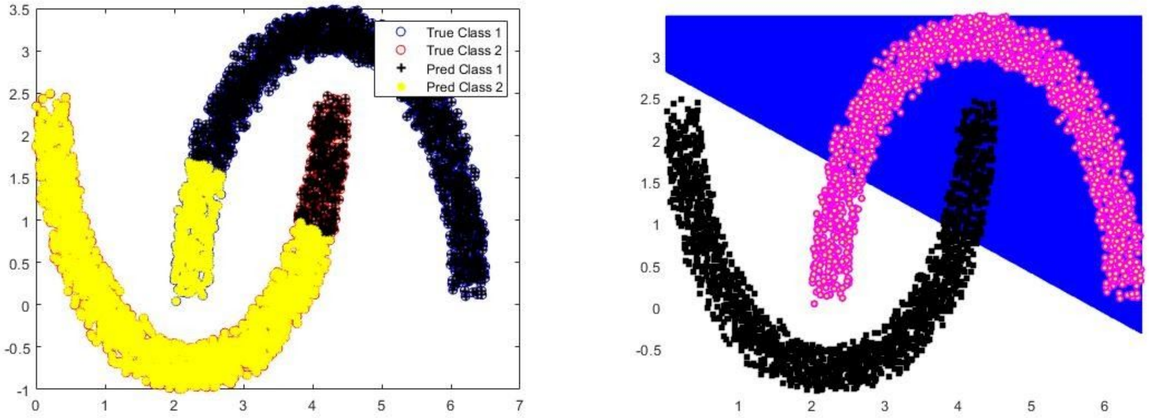
- Classification using Linear Discriminant Model:


Prediction Decision Region

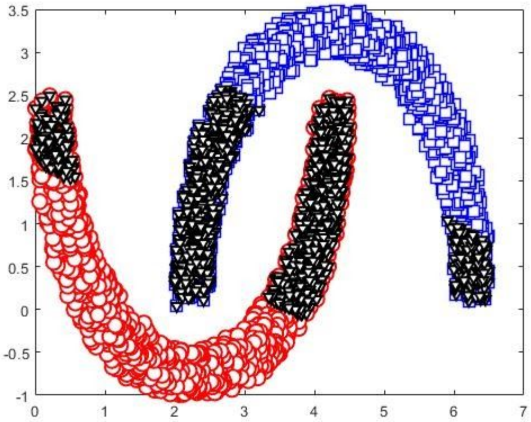
- Classification using Quadratic Discriminant Model:


Prediction Decision Region

- Classification using SVM*[4]* Linear Model:

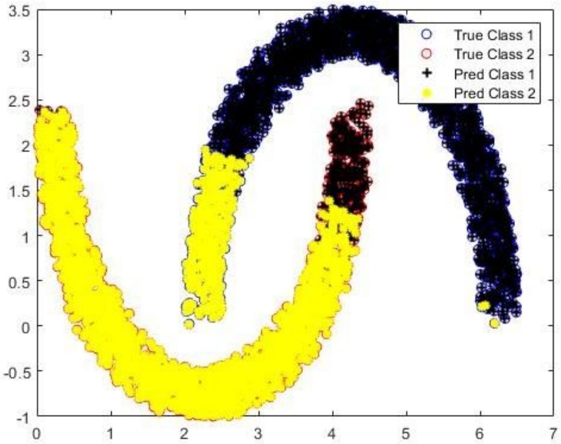

Prediction Decision Region

Support Vectors *[5]*:

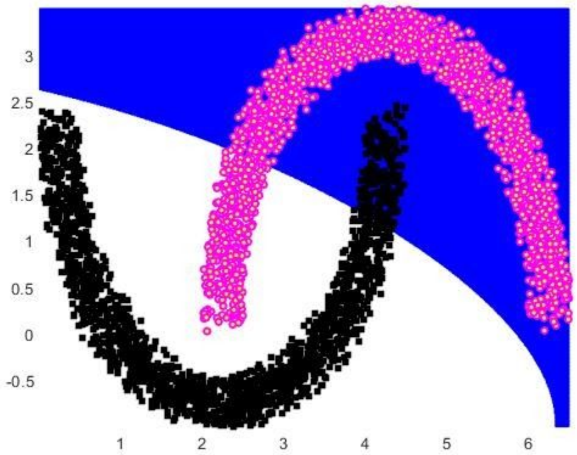


- Classification using SVM Quadratic Model:

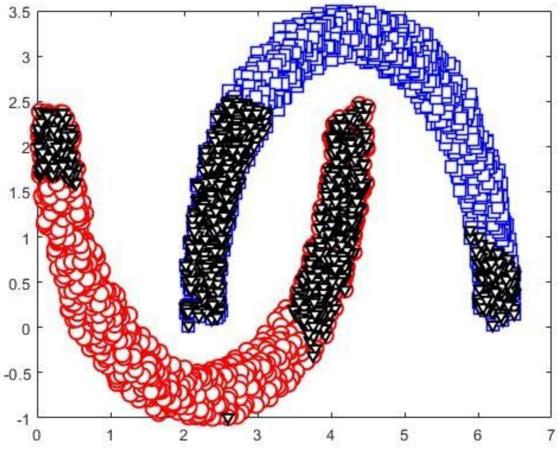
Prediction



Decision Region

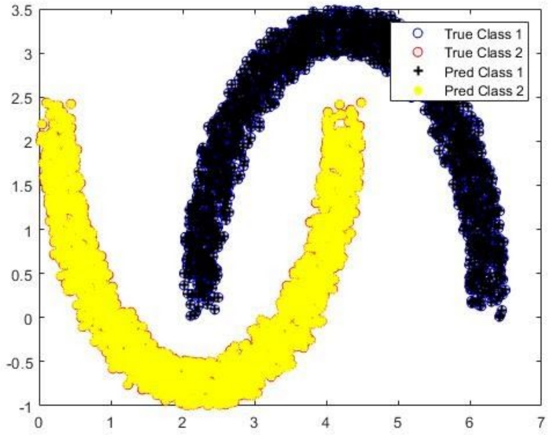


Support vectors:

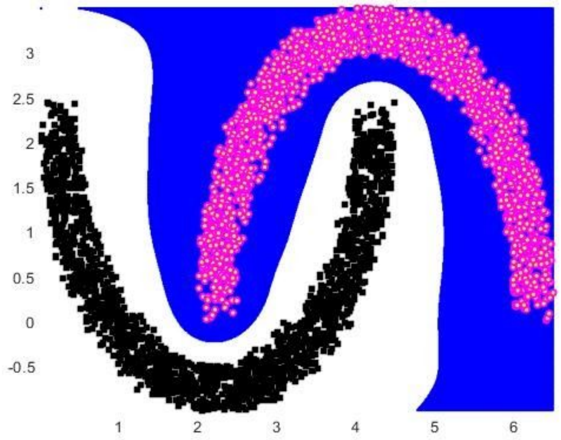


- Classification using SVM Gaussian Model:

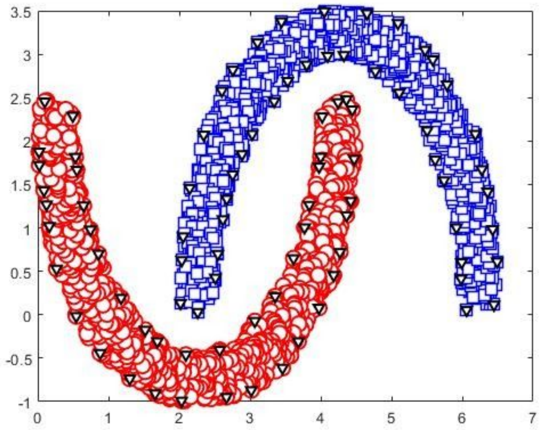
Prediction



Decision Region

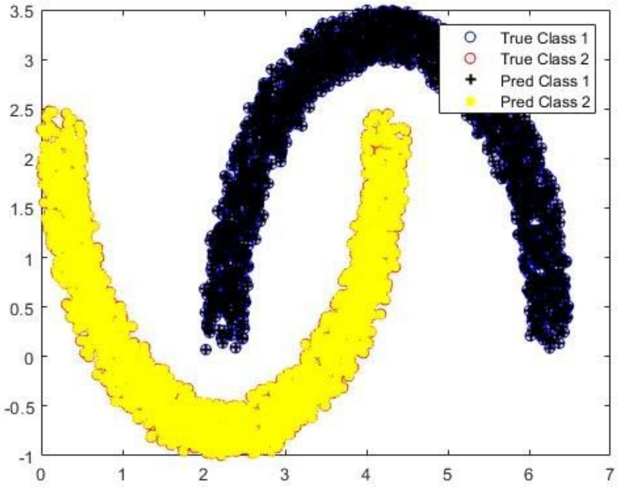


Support vectors:

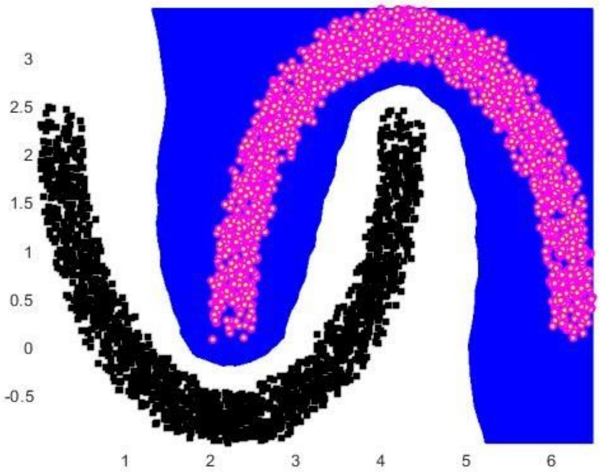


- Classification using KNN[6] Model:

Prediction



Decision Region



5. Analysis:

5.1 Problem 1:

- Confusion Matrix: [1000 2 ; 0 998]
- Accuracy : 0.9990

The accuracy is good which says that the data fits exactly in the model.

5.2 Problem 2:

Model Name	Accuracy(%)
Linear Discriminant	84.9%
Quadratic Discriminant	85.6%
Support vector machine (Linear)	84.8%
Support vector machine (Quadratic)	84.5%
Support vector machine (Gaussian)	100%
K-nearest neighbor (Fine)	100%

6. MatLab code:

Here is the drive link for matlab codes we are using drive link to optimise to this report

6.1 Matlab code for Problem 1: matlab_code _problem_01

6.2 Matlab code for Problem 2: matlab_code _problem_02

[1] Euclidean distance $d(p,q) = \sqrt{\sum_{i=1}^{n} (q_i - p_i)^2}$,

[2] The method of least squares determines the coefficients such that the **sum of the square** of the deviations between the data and the curve-fit is minimized,

[3] In the field of machine learning and specifically the problem of statistical classification, a confusion matrix, also known as an error matrix, is a specific table layout that allows visualization of the performance of an algorithm ,

[4] Support vector machine a machine learning algorithm for classification of data,

[5] support-vector machines are supervised learning models with associated learning algorithms that analyze data used for classification and regression analysis,

[6] k-nearest neighbor a machine learning algorithm for classification of data.,