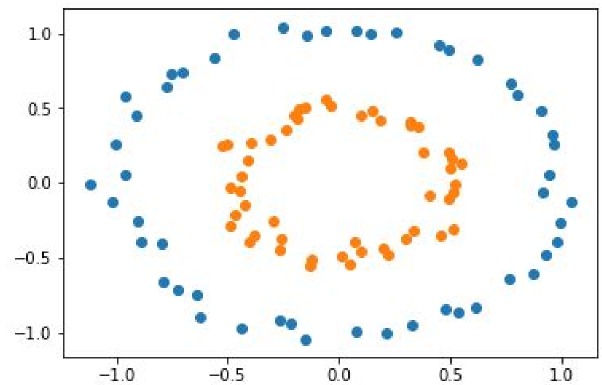
MACHINE LEARNING
Assignment 2

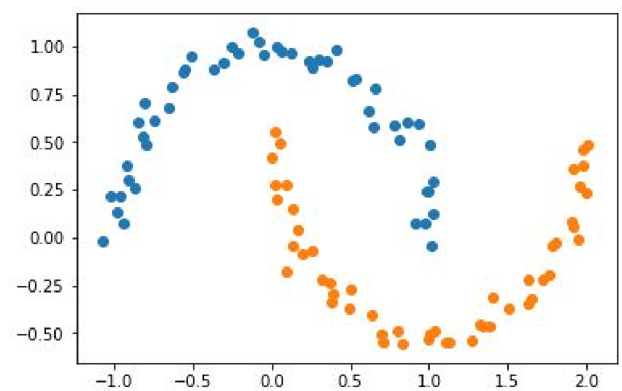
Surabhi S Nath
2016271

QUESTION 1 - Plot and observe the datasets

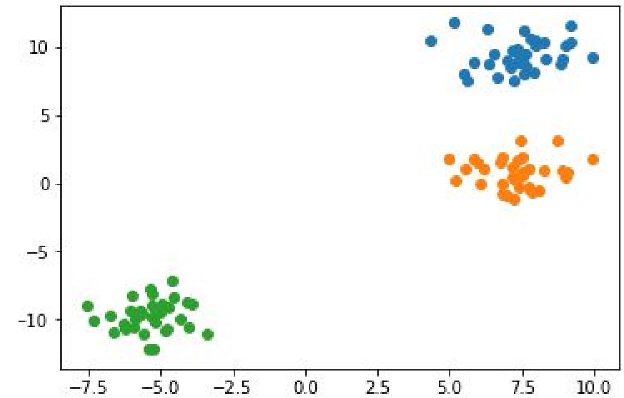
1. Dataset 1:
   Observations:
   A. Number of Samples = 100
   B. Number of Classes = 2
   C. Balance among Classes = Equally balanced (50 samples in both classes)
   D. Separability - Not linearly separable (Can be linearly separable on mapping to higher dimension)
   E. Noise in the dataset - Low noise



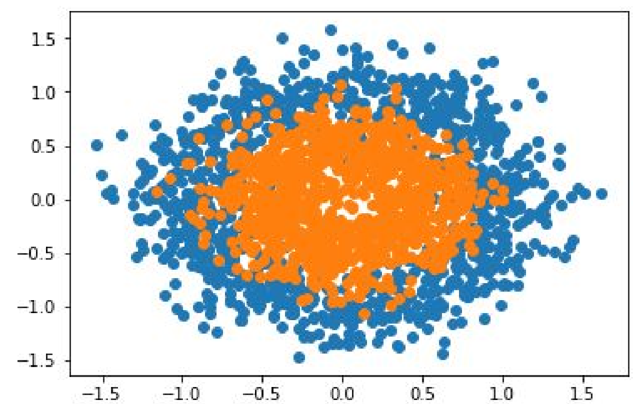
2. Dataset 2:
   Observations:
   A. Number of Samples = 100
   B. Number of Classes = 2
   C. Balance among Classes = Equally balanced (50 samples in both classes)
   D. Separability - Not linearly separable (Can be linearly separable on mapping to higher dimension)
   E. Noise in the dataset - Low noise



3. Dataset 3:
   Observations:
   A. Number of Samples = 100
   B. Number of Classes = 3
   C. Balance among Classes = Equally balanced (Around 33 samples in all classes)
   D. Separability - Linearly separable
   E. Noise in the dataset - Low noise



4. Dataset 4:
   Observations:
   A. Number of Samples = 2000
   B. Number of Classes = 2
   C. Balance among Classes = Equally balanced (1000 samples in both classes)
   D. Separability - Not linearly separable (Can be linearly separable on mapping to higher dimension)
   E. Noise in the dataset - High noise

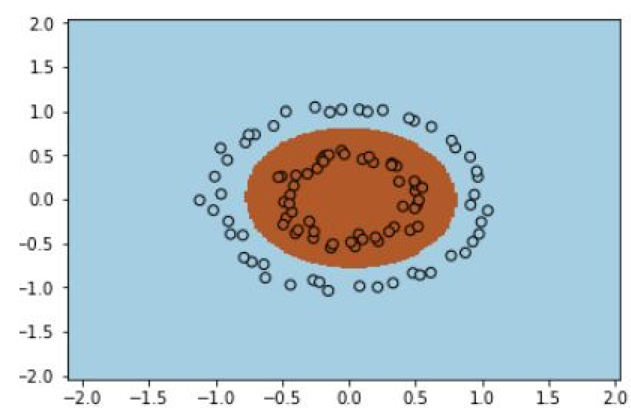


QUESTION 2 - Define Kernels and Plot decision Boundaries

1. Dataset 1:
   A. Kernel used - Polynomial Kernel with p = 2

   $$\mathbf{K(x,x') = (x.x' + 1)^2}$$

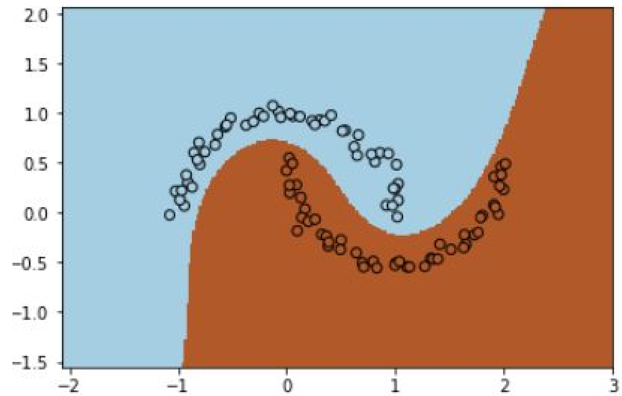
   B. Reason for choice of kernel - Since the data is present as 2 distinct concentric circles, a polynomial kernel with degree 2 can successfully separate the 2 classes



2. Dataset 2:
    A. Kernel used - Polynomial Kernel with p = 3

    $K(x,x') = (x.x' + 1)^3$

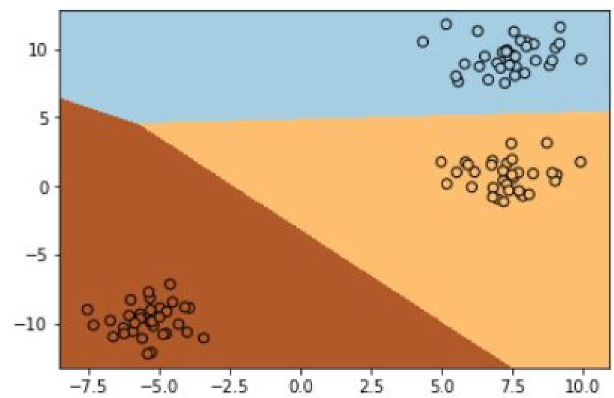
    B. Reason for choice of kernel - Since here, the data is present as 2 half moons, a cubic kernel with degree 3 can successfully separate the 2 classes



3. Dataset 3:
    A. Kernel used - Linear Kernel

    $K(x,x') = x.x'$

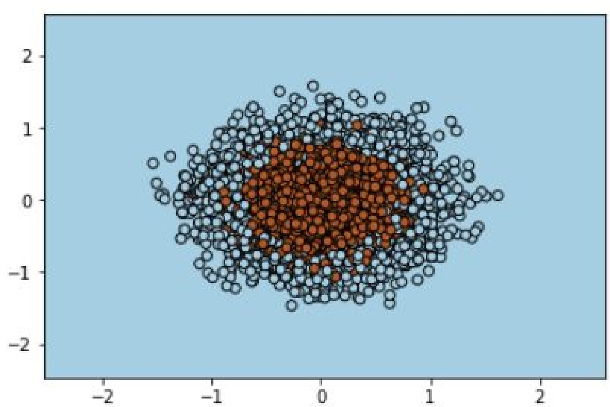
    B. Reason for choice of kernel - Since here, the data is present 3 small groups, a linear boundary can easily separate the 3 classes



4. Dataset 4:
    A. Kernel used - Polynomial Kernel with p = 2

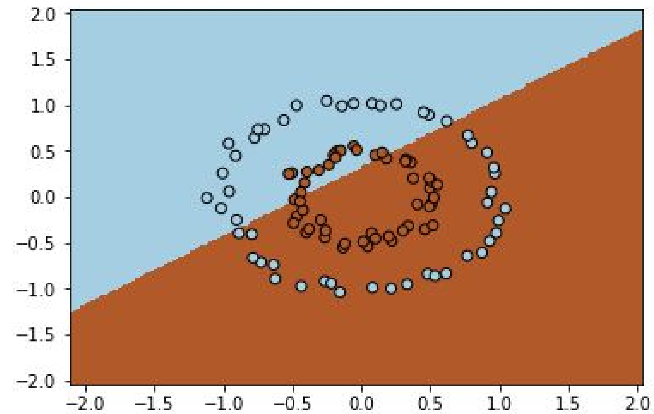
    $K(x,x') = (x.x' + 1)^2$

    B. Reason for choice of kernel - Since here, the data is present as a ring and a circle within the ring, a polynomial kernel with degree 2 can successfully separate the 2 classes, although, a soft margin will be adopted in order to classify the noisy data

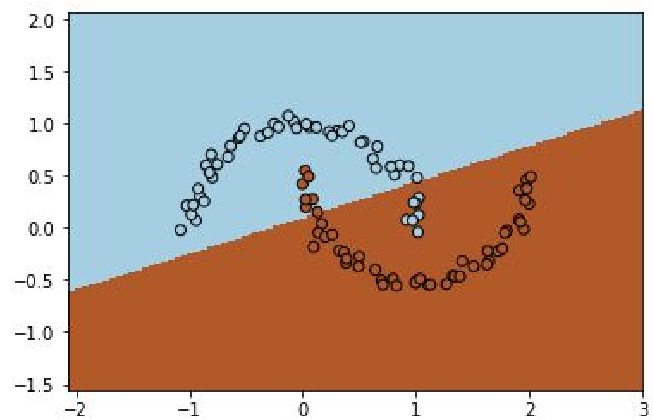


QUESTION 3 - Soft margin SVM using Linear Kernel

1. Dataset 1:
    A. Accuracy:
        a. Training - 0.6
        b. Testing - 0.54
    B. F1 Score:
        a. For class 1 - 0.44444444
        b. For class 2 - 0.54545455

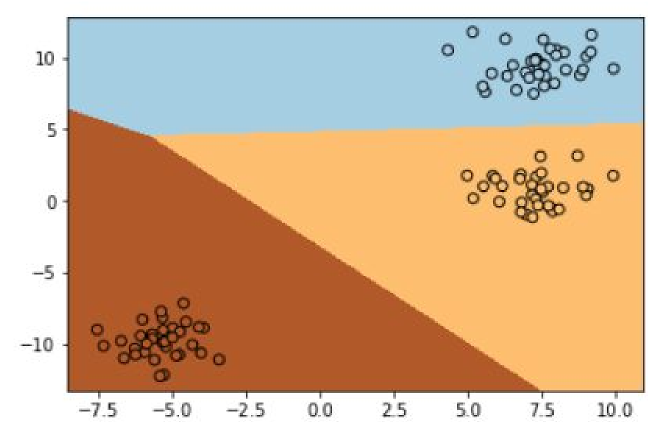


2. Dataset 2:
    A. Accuracy:
        a. Training - 0.85
        b. Testing - 0.84
    B. F1 Score:
        a. For class 1 - 0.8
        b. For class 2 - 0.88

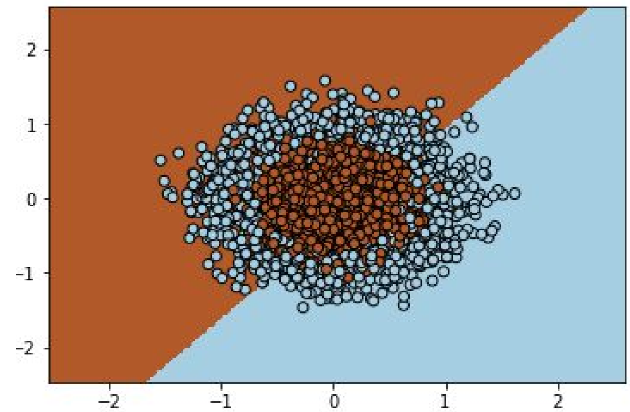


3. Dataset 3:
    ONE VS REST
    A. Accuracy:
        a. Training - 0.85
        b. Testing - 0.84
    B. F1 Score:
        a. For class 1 - 1
        b. For class 2 - 1
        c. For class 3 - 1

    ONE VS REST
    C. Accuracy:
        a. Training - 0.85
        b. Testing - 0.84
    D. F1 Score:
        a. For class 1 - 1
        b. For class 2 - 1
        c. For class 3 - 1

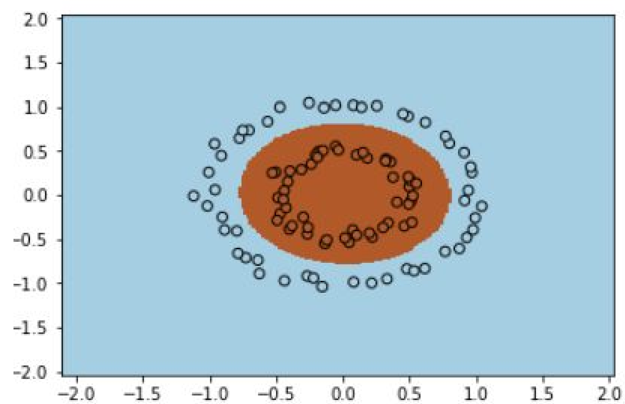


4. Dataset 4:
    A. Accuracy:
        a. Training - 0.509
        b. Testing - 0.5
    B. F1 Score:
        a. For class 1 - 0.44514107
        b. For class 2 - 0.63201663

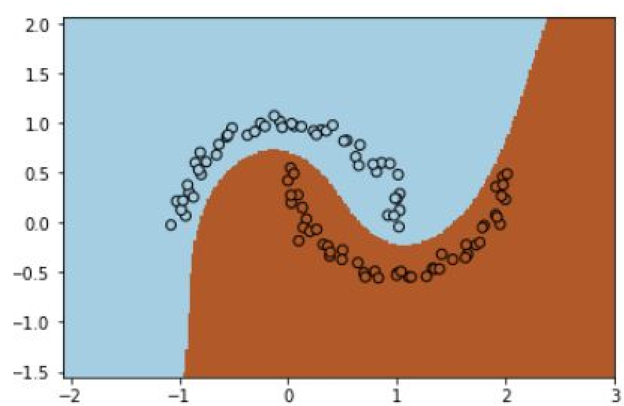


QUESTION 4 - Soft margin SVM using RBF Kernel

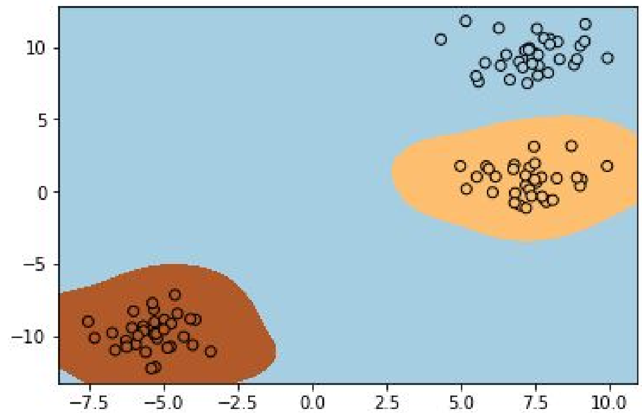
1. Dataset 1:
    A. Accuracy:
        a. Training - 1.0
        b. Testing - 1.0
    B. F1 Score:
        a. For class 1 - 1
        b. For class 2 - 1



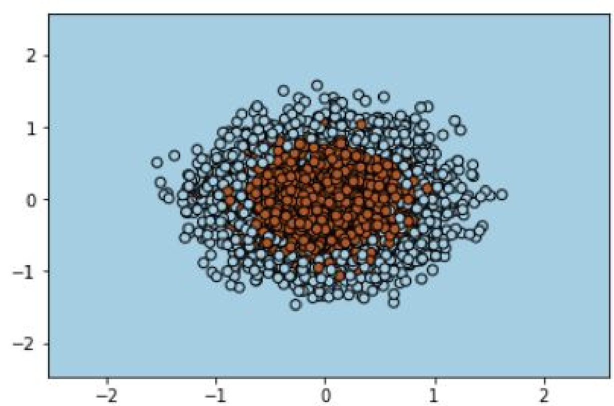
2. Dataset 2:
    A. Accuracy:
        a. Training - 1.0
        b. Testing - 1.0
    B. F1 Score:
        a. For class 1 - 1
        b. For class 2 - 1



3. Dataset 3:
    ONE VS REST
    A. Accuracy:
        a. Training - 1.0
        b. Testing - 1.0
    B. F1 Score:
        a. For class 1 - 1
        b. For class 2 - 1
        c. For class 3 - 1
    ONE VS ALL
    A. Accuracy:
        a. Training - 1.0
        b. Testing - 1.0
    B. F1 Score:
        a. For class 1 - 1
        b. For class 2 - 1
        c. For class 3 - 1



4. Dataset 4:
    A. Accuracy:
        a. Training - 0.884375
        b. Testing - 0.888
    B. F1 Score:
        a. For class 1 - 0.85714286
        b. For class 2 - 0.84754522



QUESTION 5 - Hindi Character Dataset

5 classes of Data - 0, 1, 2, 3, 4 - ka, kha, ga, gha, nga

- Number of train samples = 8500
- Number of train samples per class = 1700
- Number of test samples = 1500
- Number of test samples per class = 300

Performed Classification using SVM, RBF kernel, applied GridSearchCV to determine C and Gamma. Applied 5 fold Cross Validation.

Gamma values = [0.001,0.01,0.1,0.5,1]
C values = [0.1,0.3,1,2.5,5,10,50,100,500,1000]

Best C = 2.5
Best Gamma = 0.001

Mean training accuracy over all folds - 0.978
Mean validation accuracy over all folds - 0.87552941
Test accuracy = 0.9706666666666667

QUESTION 6

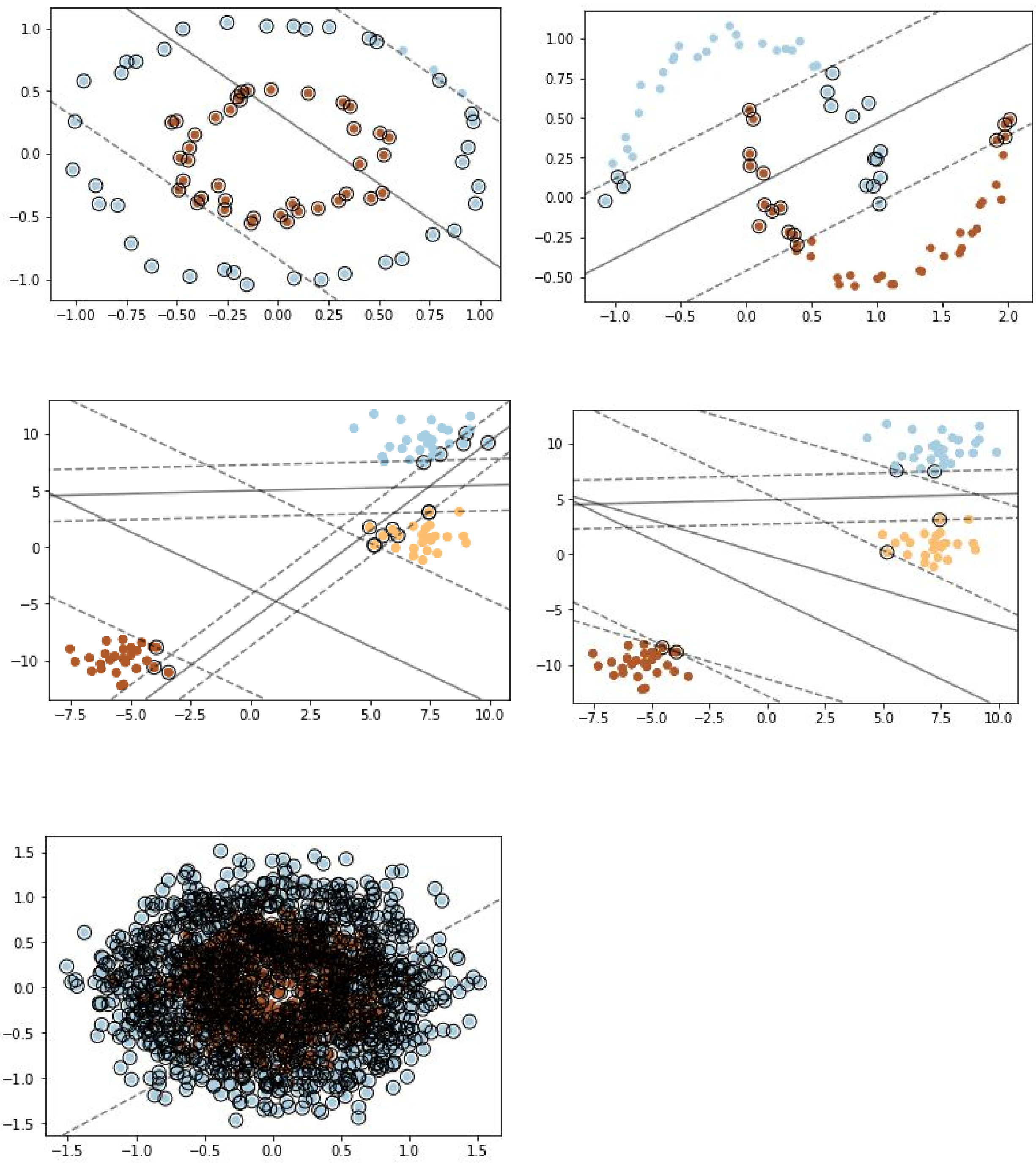
1. What hyperparameters did you choose?
    A. Soft Margin SVM using Linear Kernel
        a. Dataset 1: C = 1 - Default value of 1
        b. Dataset 2: C = 0.5 - Since we are using linear kernel, want to allow some misclassifications to obtain better decision boundary
        c. Dataset 3: C = 0.1 - Want to allow misclassification
        d. Dataset 4: C = 0.5 - Since data has noise, need a softer margin hence lower C

    B. Soft Margin SVM using RBF Kernel
        a. Dataset 1: C = 1 - Default value of 1
        b. Dataset 2: C = 10 - Since want lesser misclassifications
        c. Dataset 3: C = 0.5 - Want to allow some misclassification
        d. Dataset 4: C = 0.5 - Since data has noise, need a softer margin hence lower C

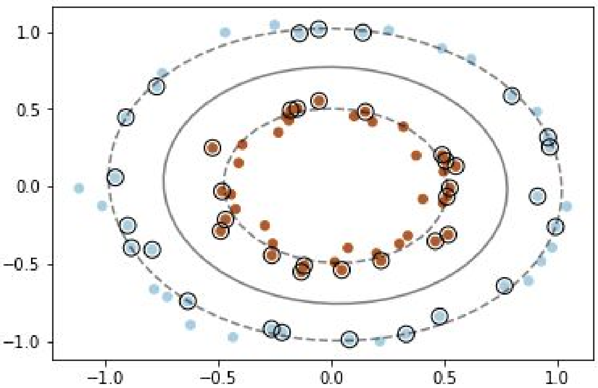
    C. Hindi Character Dataset
    GridSearchCV was used and best parameters were found to be Gamma = 0.001 and C = 2.5. The Gamma value should be around 1/number of features and since the images have 1024 (32 x 32) features, gamma = 0.001.

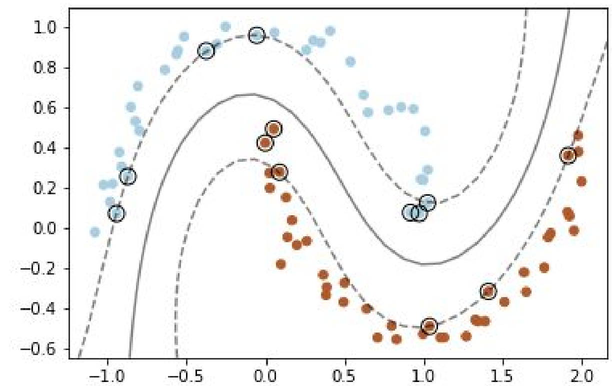
2. Plots of Support Vectors

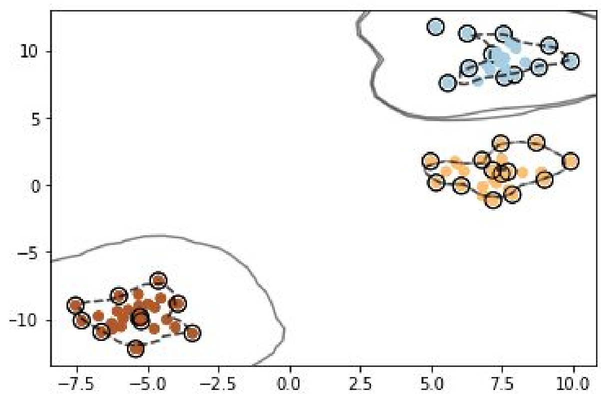
A. Soft Margin SVM using Linear Kernel

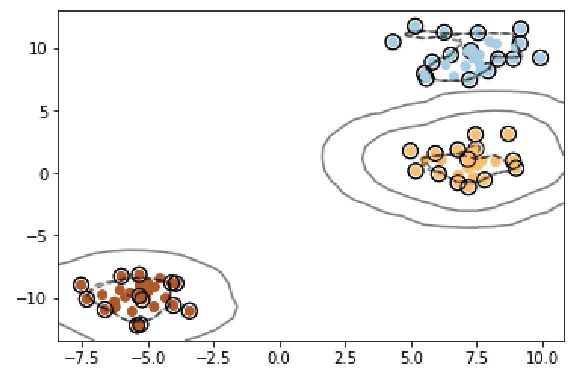


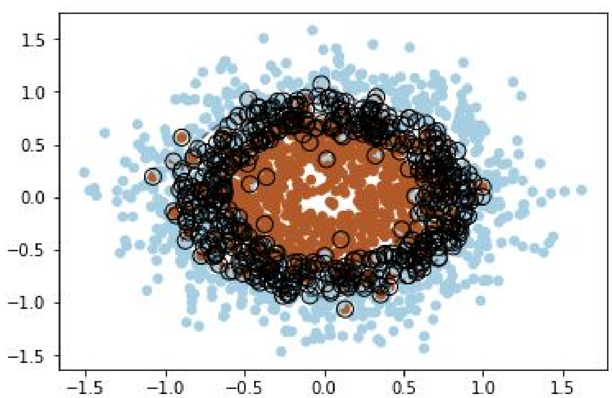
B. Soft Margin SVM using RBF Kernel


1.0
0.5
0.0
−0.5
−1.0
−1.0 −0.5 0.0 0.5 1.0

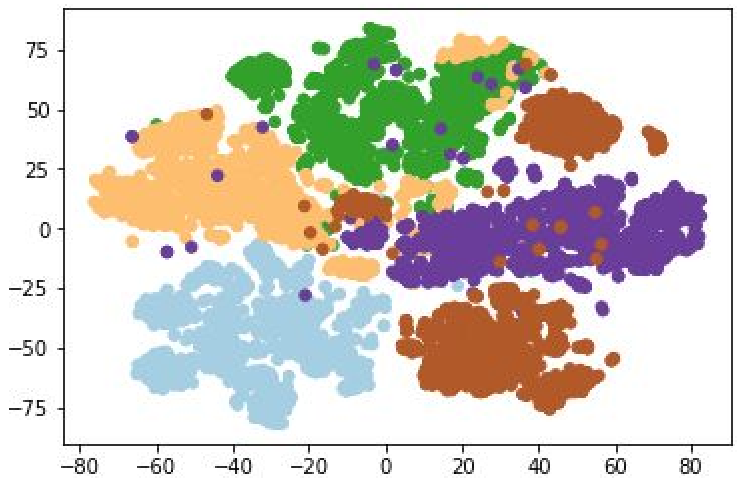

1.0
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6
−1.0 −0.5 0.0 0.5 1.0 1.5 2.0


10
5
0
−5
−10
−7.5 −5.0 −2.5 0.0 2.5 5.0 7.5 10.0


10
5
0
−5
−10
−7.5 −5.0 −2.5 0.0 2.5 5.0 7.5 10.0


1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−1.5 −1.0 −0.5 0.0 0.5 1.0 1.5

C. Hindi Character Dataset



3. Overfitting

Since the datasets can be separated using polynomial kernel, when we use a linear kernel, the model always underfits on the data. Hence, no matter what hyperparameters we choose, a linear decision boundary cannot over fit a complex data distribution such as concentric circles.

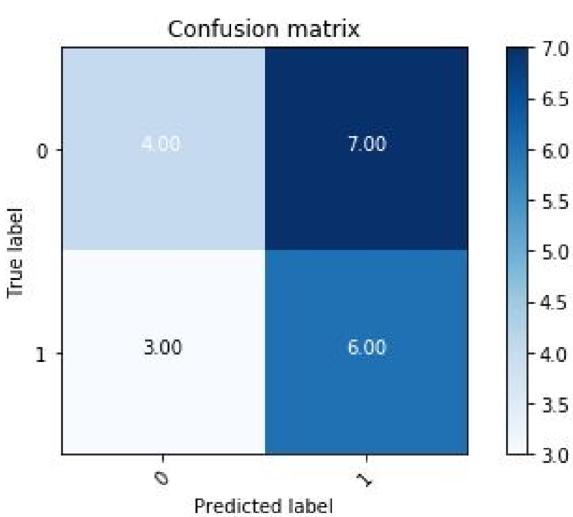
While using the RBF kernel, since dataset 1, 2 and 3 have very little noise, a clear decision boundary can be obtained for all. Since we are not using cross validation or any other form of extensive training, overfitting does not occur noticeably even after increasing C. In the 4th dataset, perhaps having a large value of gamma may overfit the data but even though the data is noisy, it still defines a circular decision boundary hence the model cannot overfit the data to reduce test accuracy greatly.

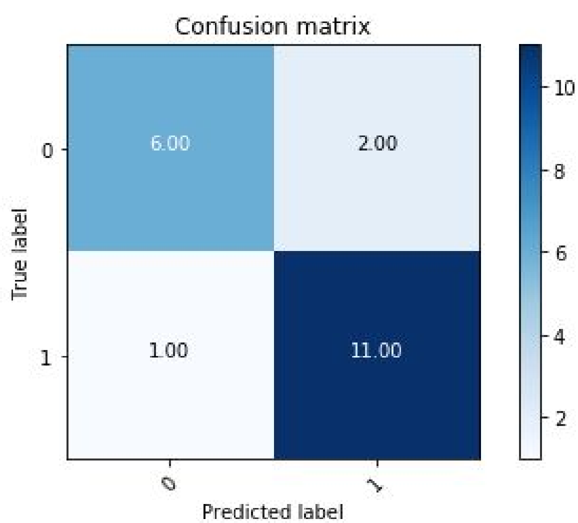
Since the equation of the decision boundary for an SVM can be found only with the support vectors, more number of support vectors indicate a more complex equation. We know that overfitting leads to a complex decision boundary. Hence, if the data overfits, we obtain more support vectors.

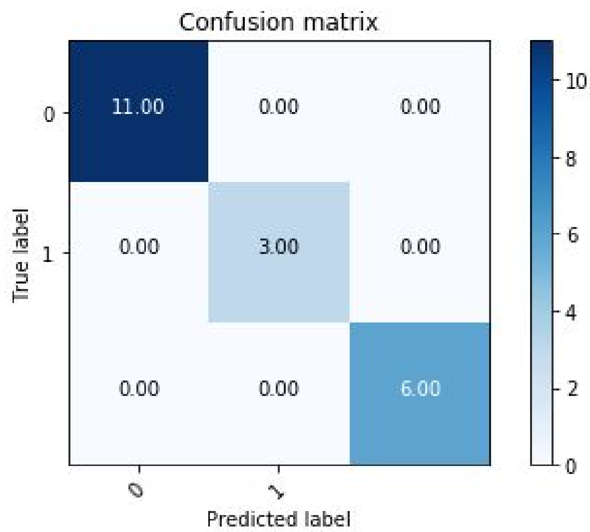
4. Linear vs RBF

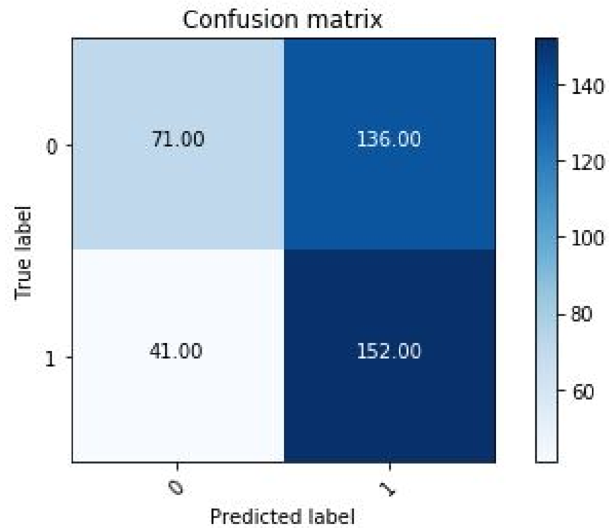
    A. Linear kernel did not perform well on the datasets 1, 2, 4 since they are not linearly separable. It worked well for dataset 3 since we cana clearly draw lines to separate the 3 classes in the data. Hence we conclude that linear kernel may often underfit complex data and not yield good accuracies.
    B. The RBF kernel performed well for all datasets. It gave a 100% accuracy in datasets 1, 2 and 3 since a clear separating decision boundary was achieved. In the 4th dataset however, due to a significant amount of noise, the RBF kernel could not yield the best accuracy. We conclude that RBF kernels are more suitable for complex data.

5. Confusion Matrices
    A. Soft Margin SVM using Linear Kernel

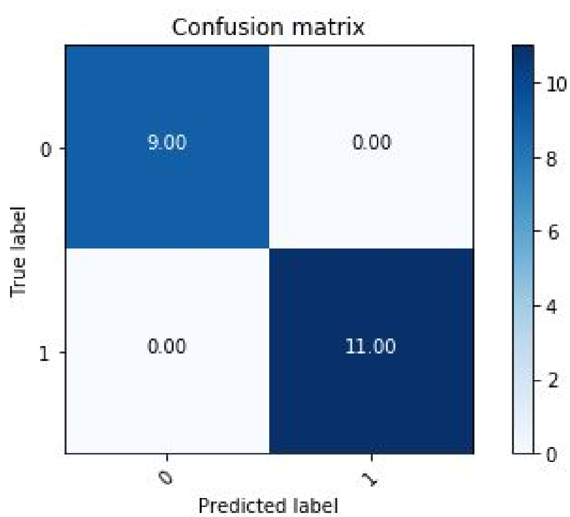


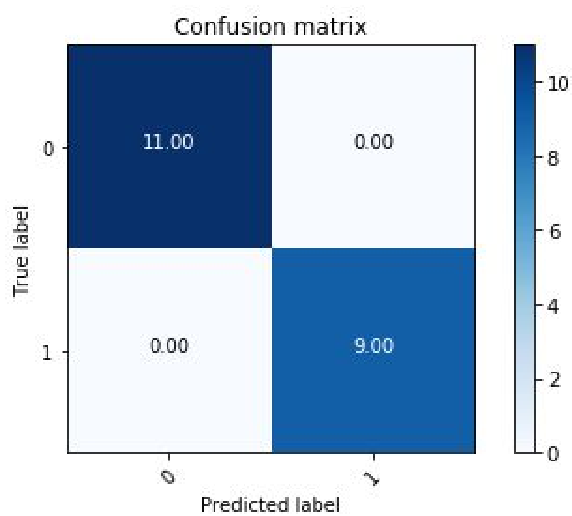


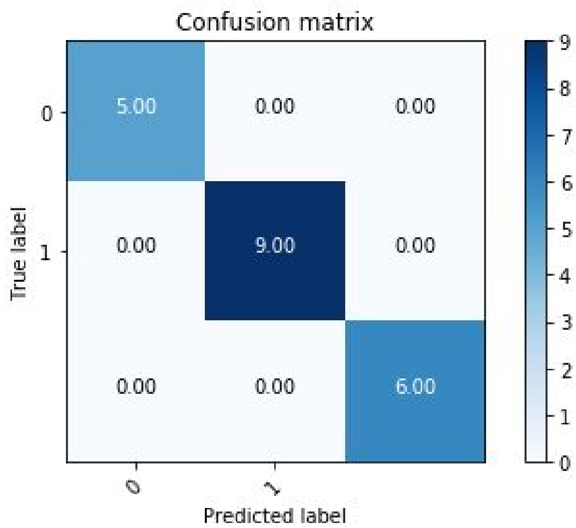


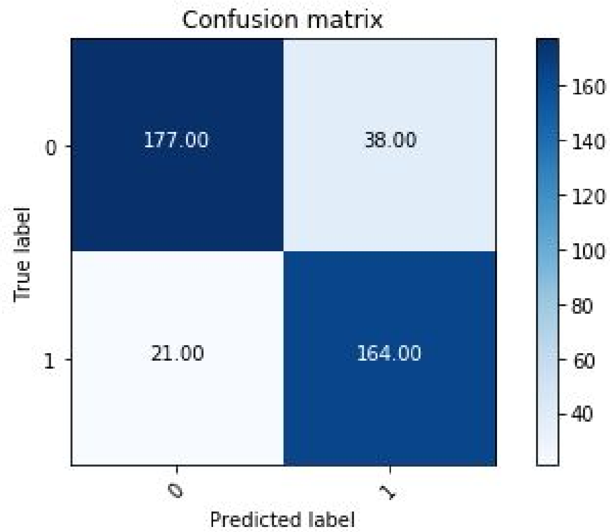


B. Soft Margin SVM using RBF Kernel

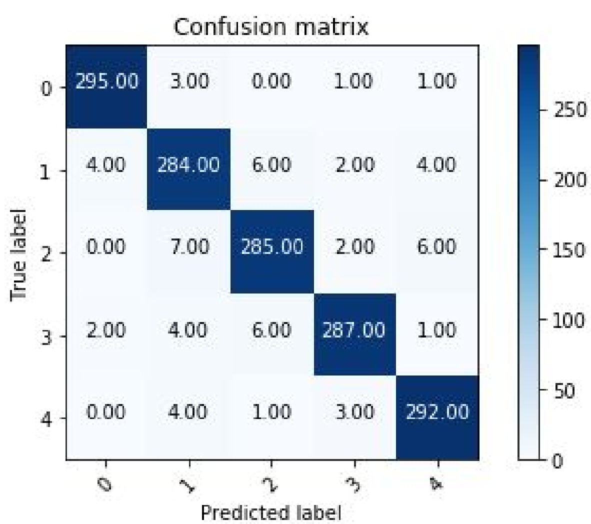






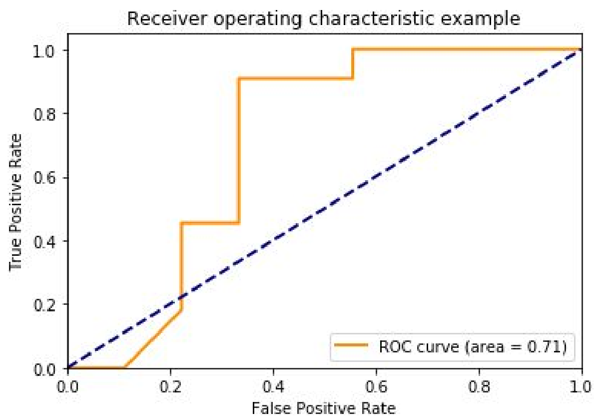


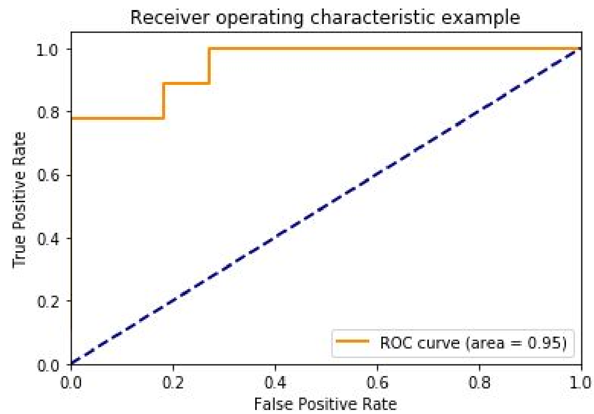
C. Hindi Character Dataset

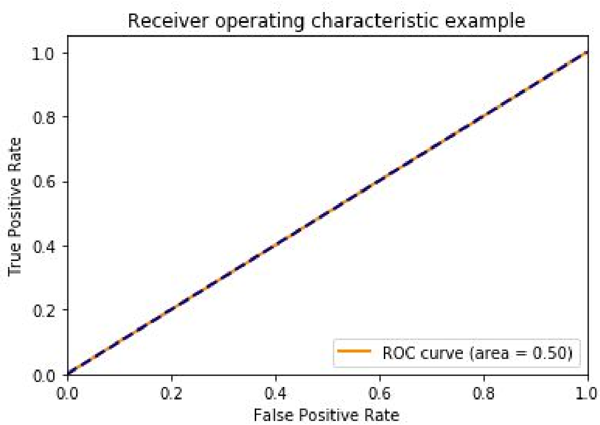


6. ROC Curves

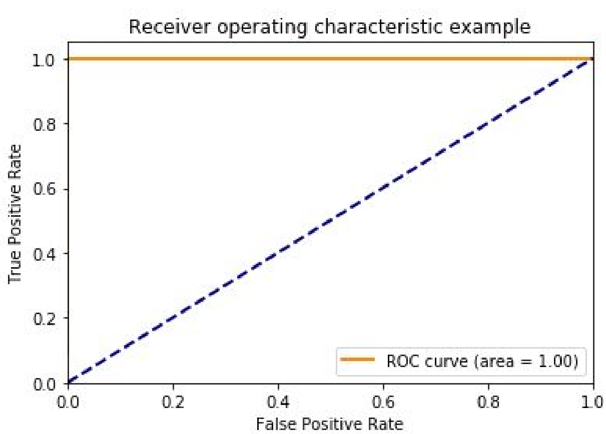
A. Soft Margin SVM using Linear Kernel

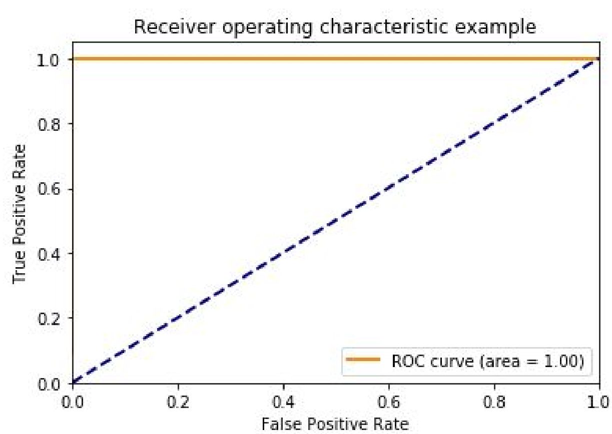


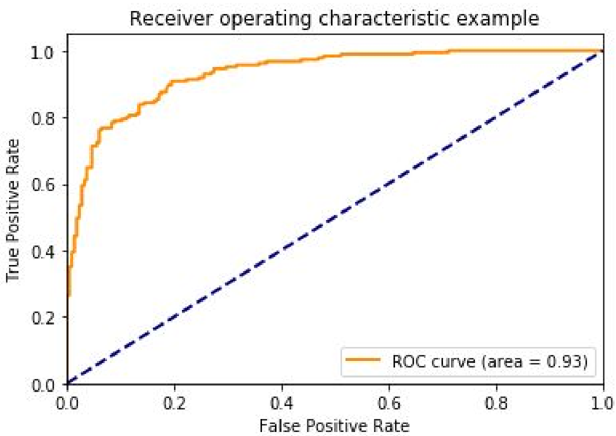




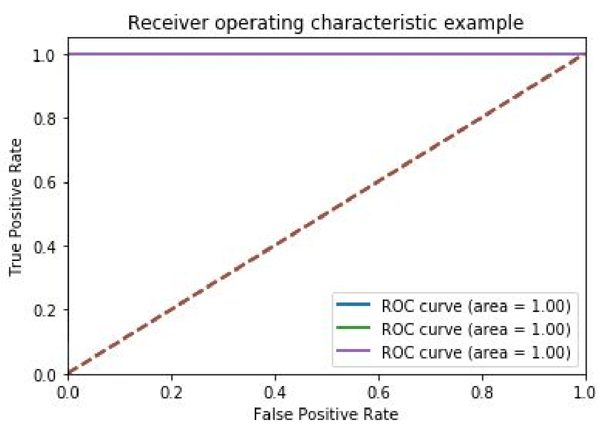
B. Soft Margin SVM using RBF Kernel

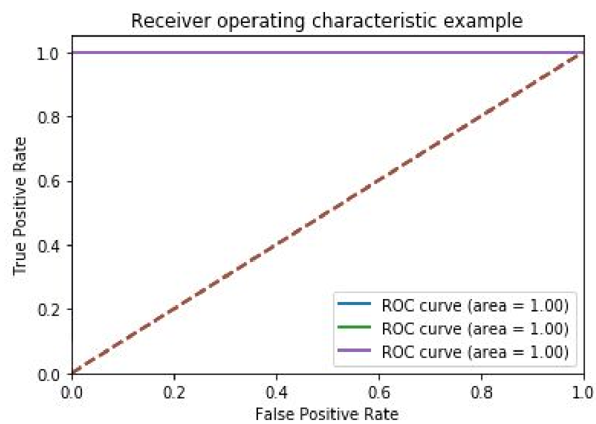




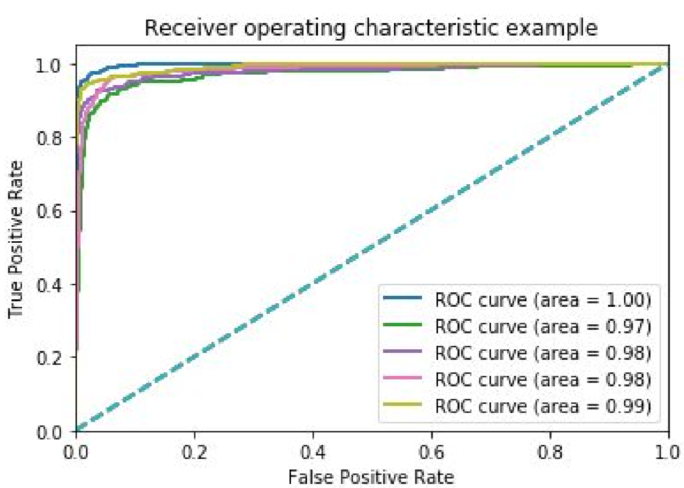


C. Bonus - Multiclass ROC for dataset 3 (linear, RBF)





D. Hindi Character Dataset



Functions Implemented:

1. Kernel1(x,y) - Implements Linear Kernel
2. Kernel2(x,y,p=2) - Implements Polynomial Kernel with degree 2
3. Kernel3(x,y,p=3) - Implements Polynomial Kernel with degree 3
4. Calc_acccnt,y_train,predictions_train,y_test,predictions_test) - Calculates accuracy
5. Binclf_linear(c) - Implements Linear Kernel SVM for binary labels
6. Binclf_rbf(c) - Implements RBF Kernel SVM for binary labels
7. Ovr_linear(num_classes, c, x_train, x_test, y_train, y_test) - Implements One vs Rest Linear SVM for multiclass labels
8. Ovo_linear(num_classes, c, x_train, x_test, y_train, y_test) - Implements One vs One Linear SVM for multiclass labels
9. Ovr_rbf(num_classes, c, x_train, x_test, y_train, y_test) - Implements One vs Rest RBF SVM for multiclass labels
10. Ovo_rbf(num_classes, c, x_train, x_test, y_train, y_test) - Implements One vs One RBF SVM for multiclass labels
11. Roc - Calculates AUC
12. Plot_roc - Plots ROC curve for binary labels
13. multiclass(X, Y, num_classes,s) - Plots ROC curve for multiclass labels
14. plot(x_train,y_train,clf) - Plots SVM decision boundary with Support Vectors
15. make_confmat(y_pred, y_act,num_classes) - Returns Confusion matrix

---