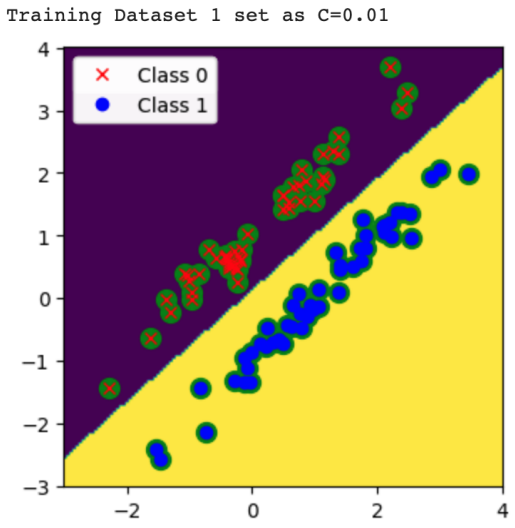
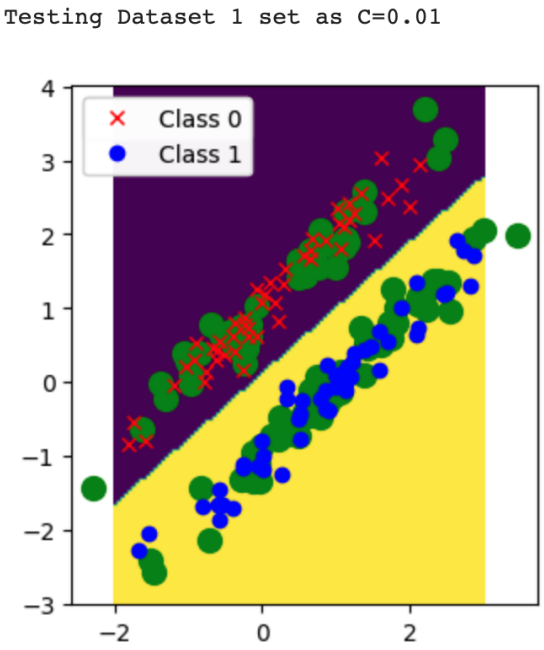
1.

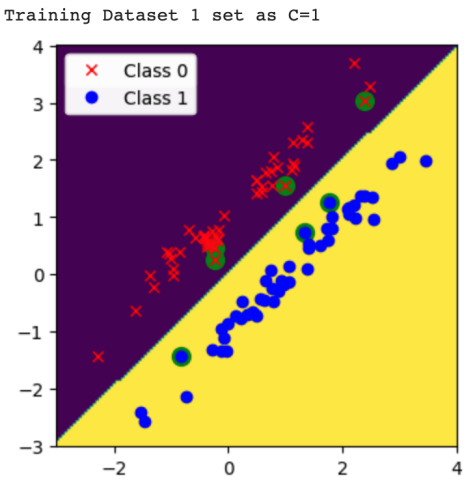
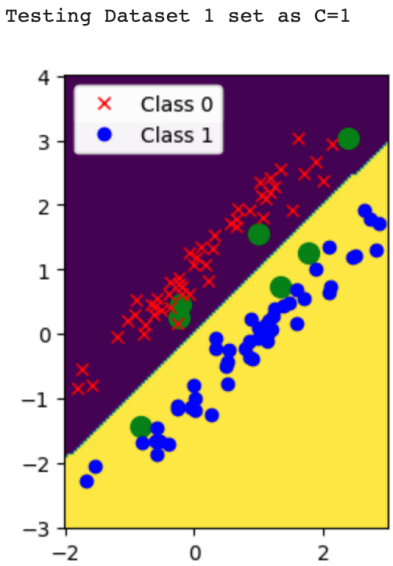
What is the meaning of parameter C and how will it impact your classification? Set C = 0.01 and C = 1. Report the above items and also provide the support vectors in the plots for each value of C. Discuss your results and explain the performance and the differences for the different values of C

a) The slack parameter C is the regularization parameter. The parameter C when we increase it affects the classification by decreasing the distance of the margin from the boundary. When C = 0.01 the distance of the margin from the boundary appears to be large so the points in this region dont give any error in the criterion error. Because our goal is to have a smaller margin region, it is a better choice to have a larger C value (for example when C=1). When C = 1, it can be seen from the graphs that the majority of the points were outside and there were less than 5 points inside the margin, so it has a better classification.

When C = 0.01

	
Training Accuracy is 100.0% when C=0.01 Testing Accuracy is 100.0% when C=0.01	w (weight vector)=[0.47617753 -0.53480959] w0 (offset)=0.0636915652500464

When C = 1

	
Training Accuracy is 100.0% when C=1 Testing Accuracy is 100.0% when C=1	w (weight vector)=[1.52672792 -1.54421194] w0 (offset)=0.04306309183346128

Support Vectors

C = 0.01	C = 1
Support Vectors = [[2.19929132 3.69039929] [-0.22111103 0.66801655] [0.49293496 1.63599988] [2.38349209 3.03218741] [1.13712786 2.31828414] [-1.02073547 0.26766422] [-0.56729058 0.64096728] [0.59560878 1.52144654] [-0.22862554 0.25995578] [-0.68307677 0.77811072] [-1.31467374 -0.2284476] [-1.36838283 -0.02790532] [-0.27512966 0.74678986] [-0.86053448 0.39736139] [-0.17144342 0.76899905] [-0.42714834 0.68511456] [0.5008731 1.4157754] [0.49579778 1.65522892] [0.79924873 2.05736087] [0.75268513 1.80975798] [0.87039876 1.86493455] [-0.31593348 0.5154707] [1.37393231 2.58265345] [-0.17934762 0.62860895]	Support Vectors = [[2.38349209 3.03218741] [-0.22862554 0.25995578] [-0.22392623 0.45364466] [1.00356818 1.55545471] [-0.8194963 -1.42969518] [1.77344668 1.25547895] [1.34639848 0.73273639]]

```
[-0.25017869  0.49434588]
[ 0.64773632  1.78474002]
[-0.22392623  0.45364466]
[-0.40753748  0.64388851]
[ 1.15297686  1.89907387]
[-0.37872232  0.49229385]
[-0.36587061  0.52632341]
[-2.29296754 -1.44205184]
[-0.9569205   0.1021167 ]
[-0.43504242  0.66651871]
[-1.06691945  0.38613777]
[ 1.38531339  2.30181176]
[-0.19791072  0.60032311]
[-1.63120437 -0.64462429]
[ 1.10044716  1.82152393]
[-0.08573927  1.02746492]
[-0.96320673 -0.02791502]
[ 1.28617167  2.35913755]
[ 0.77405942  1.54813285]
[-0.38088622  0.56880716]
[ 0.58540282  1.43371291]
[ 1.13416337  1.95359901]
[-0.39193042  0.63106428]
[ 2.47383721  3.28301718]
[ 1.00356818  1.55545471]
[-1.03611366  0.37502682]
[ 0.47263449 -0.70927065]
[-0.02666208 -1.33582734]
[-0.11686633 -1.33634513]
[ 2.10090884  1.14377075]
[ 2.51239689  1.34597184]
[ 1.41556415  0.46556088]
[ 1.40762518  0.53703513]
[ 0.55926329 -0.4387909 ]
[ 2.32646419  1.37940582]
[ 3.00352145  2.0656142 ]
[ 0.78475411 -0.47444015]
[ 0.12405825 -0.72337497]
[ 0.97696488 -0.17817248]
[ 0.33990583 -0.70747574]
[-0.12127284 -0.96070803]
[-0.72663108 -2.14100517]
[ 0.65535204 -0.11318246]
[-0.8194963  -1.42969518]
[ 2.54049571  0.96459234]
[ 1.05473336 -0.13521068]
[ 0.74264878  0.06297634]
[-0.2875812  -1.3069237 ]
[ 1.75677702  0.59293519]
[ 1.77344668  1.25547895]
[ 0.43637164 -0.6507377 ]
[ 0.23473178 -0.4788083 ]
[ 1.81865528  1.01009377]
```

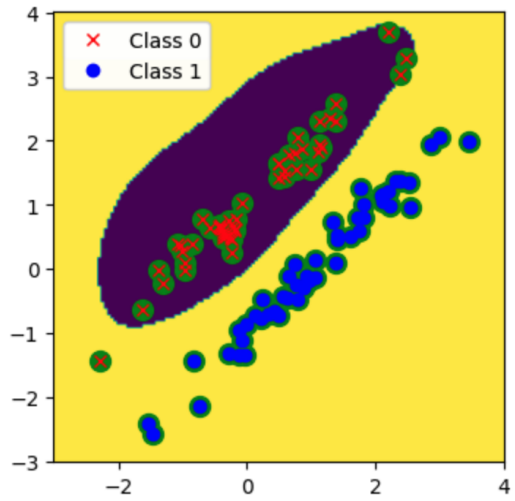
```
 [ 1.06636517  0.14695715]
 [-0.06598454 -1.11048037]
 [-1.52338298 -2.41786831]
 [ 2.12222121  1.17034602]
 [ 1.60643041  0.50440225]
 [-0.00574016 -0.86017274]
 [ 2.37873259  1.3795532 ]
 [ 2.19910636  1.20423174]
 [ 2.11149915  1.05494946]
 [-1.47522731 -2.56586838]
 [ 0.21471899 -0.78065116]
 [ 0.87718114 -0.29149344]
 [ 0.77387261 -0.24016546]
 [ 3.44764981  1.99646181]
 [ 2.22347671  0.990169  ]
 [ 0.62904188 -0.45352673]
 [ 0.49239614 -0.7355669 ]
 [ 0.91778831 -0.10176313]
 [ 2.87200337  1.94917117]
 [ 1.34639848  0.73273639]
 [ 1.81593058  0.80997629]
 [ 1.72628644  0.81140485]
 [ 1.37478245  0.10107443]]
```

Use a Gaussian (RBF) Kernel with C parameter set to C = 0.01. Set γ = 1, 3, 10, 50. Report the above items and also show the support vectors in the training-data plots for each value of γ. Explain the linearity or nonlinearity of the decision boundary, and explain the difference in decision regions for the various values of γ. State where (if anywhere) you observe underfitting or overfitting.

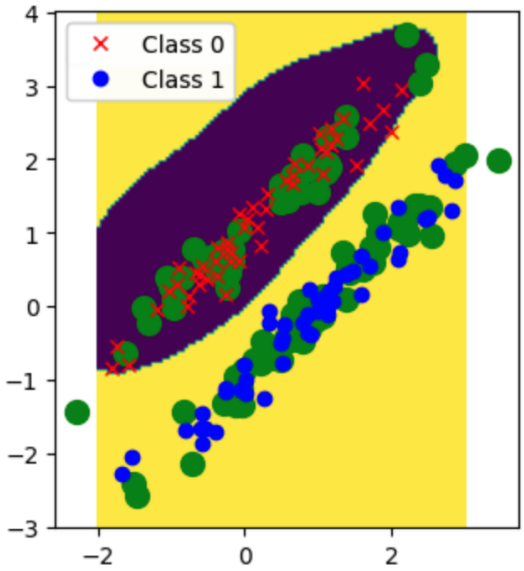
b) For all gamma values, the decision boundary is non-linear. It can be seen that as gamma increases the purple oblong shape decreases until it vanishes when gamma is 50. Another observation is that as gamma increases, overfitting can occur. When gamma is 50, you can see that the points in class 0 are on the decision boundary.

When γ = 1, C = 0.01

Training Dataset 1 set as gamma=1



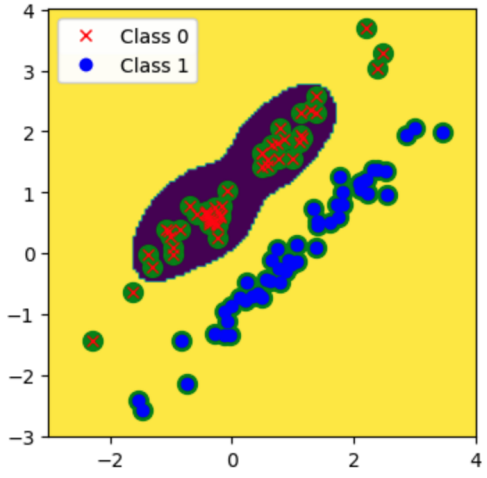
Testing Dataset 1 set as gamma=1



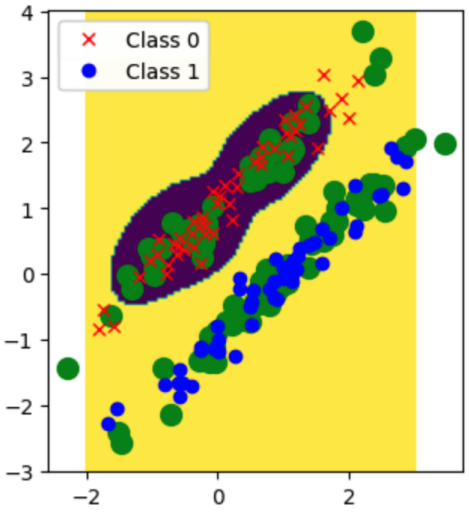
Training Accuracy is 99.0% when gamma=1
Testing Accuracy is 99.0% when gamma=1

When $\gamma = 3$, $C = 0.01$

Training Dataset 1 set as gamma=3

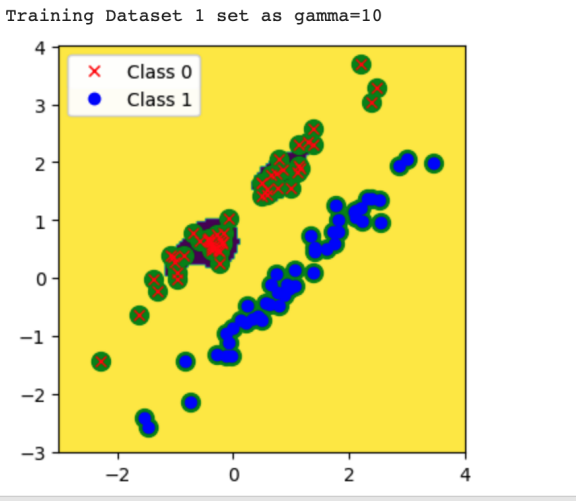


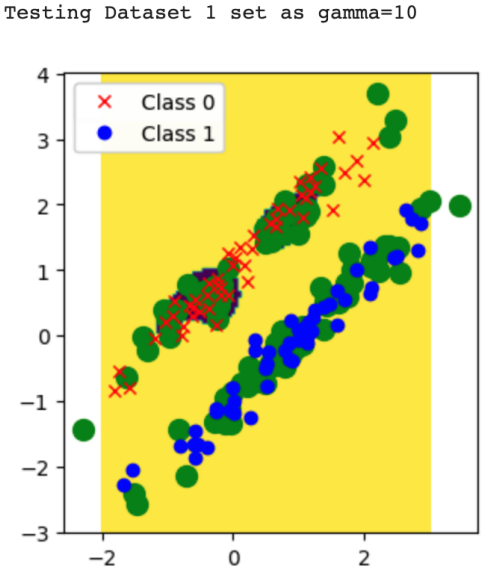
Testing Dataset 1 set as gamma=3



Training Accuracy is 95.0% when gamma=3
Testing Accuracy is 92.0% when gamma=3

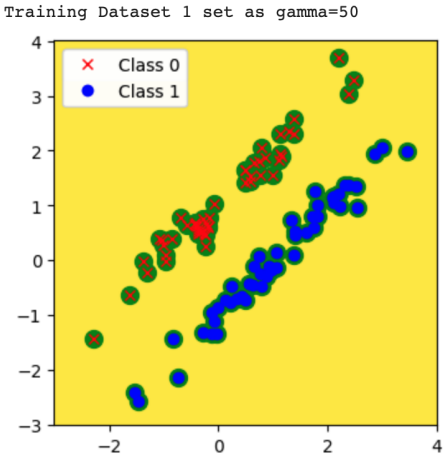
When γ = 10, C = 0.01

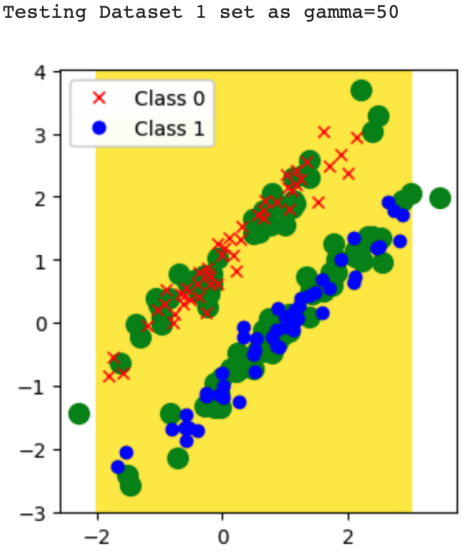




Training Accuracy is 88.0% when gamma=10
Testing Accuracy is 77.0% when gamma=10

When γ = 50, C = 0.01



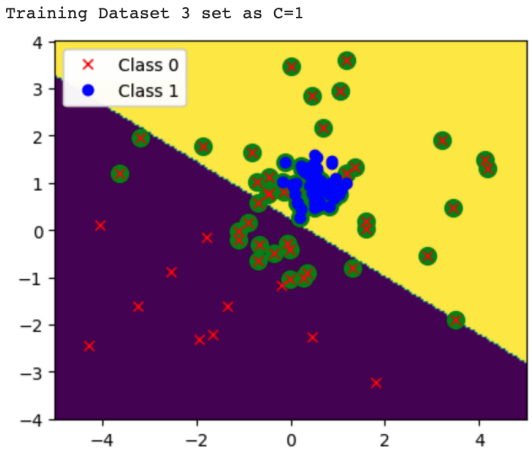
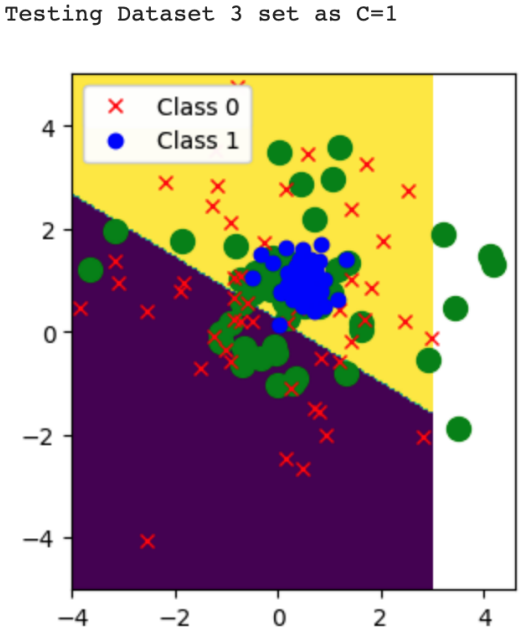


Training Accuracy is 79.0% when gamma=50 Testing Accuracy is 60.0% when gamma=50	

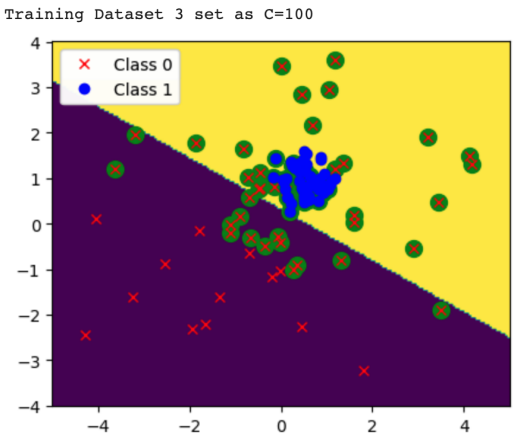
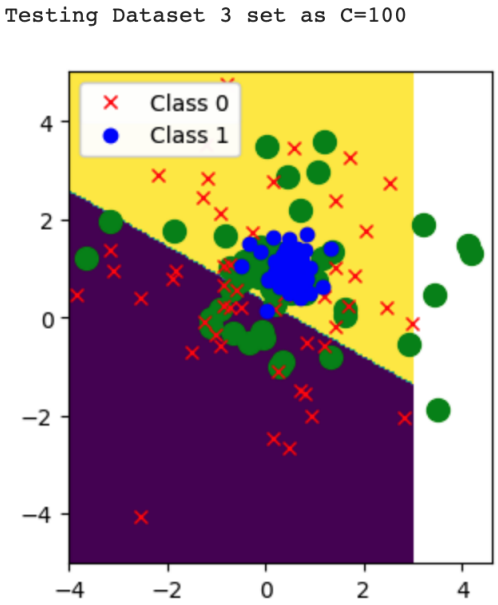
Use the Linear Kernel and try different values of slack variable parameter C. Set C = 1 and C = 100. Report the above items for each value of C. Discuss your results. You will provide 4 plots in total.

c) When increasing the value of C, the testing and training accuracy increased by 1% so there is an improvement. When C is smaller, for example 1, more points were in the decision boundary than when it is bigger like 100.

When C = 1

	
Training Accuracy is 76.0% when C=1 Testing Accuracy is 74.0% when C=1	w (weight vector)=[0.47927439 0.78667459] w0 (offset)=-0.16998607354185055

When C = 100

	
Training Accuracy is 77.0% when C=100 Testing Accuracy is 75.0% when C=100	w (weight vector)=[0.49088855 0.86936408] w0 (offset)=-0.2641121309436542

C = 1	C = 100
Support Vectors = [[-3.18218116 1.96644083] [0.33801478 -0.90621088] [1.17837711 3.60142806] [1.04601153 2.95951526] [3.48719892 -1.88711189] [0.27257033 1.32269704] [1.17347805 1.21187743] [-0.48753857 0.76858642] [-0.14786511 0.8244982] [-0.68079504 0.57298126] [-1.10325263 -0.01428632] [0.69860778 2.17485997] [0.45597766 2.85726592] [4.18590014 1.3103195] [1.60690307 0.03458257] [-0.02832969 -0.420729] [-0.71754165 1.01085467] [-1.86498157 1.77056467] [0.27889619 -1.0043282] [-0.68281116 -0.63908943]	Support Vectors = [[-3.18218116 1.96644083] [0.33801478 -0.90621088] [1.17837711 3.60142806] [1.04601153 2.95951526] [3.48719892 -1.88711189] [0.27257033 1.32269704] [1.17347805 1.21187743] [-0.48753857 0.76858642] [-0.14786511 0.8244982] [-0.68079504 0.57298126] [-1.10325263 -0.01428632] [0.69860778 2.17485997] [0.45597766 2.85726592] [4.18590014 1.3103195] [1.60690307 0.03458257] [-0.02832969 -0.420729] [-0.71754165 1.01085467] [-1.86498157 1.77056467] [0.27889619 -1.0043282] [1.60139977 0.17776224]

```
[ 1.60139977  0.17776224]
[-0.45506368  0.79536934]
[-0.44720807  1.13514243]
[ 3.20775231  1.91160833]
[ 1.35240975  1.32747597]
[-1.10754848 -0.19512252]
[-3.64007736  1.20889011]
[ 4.12858298  1.47898401]
[ 0.01816077  3.47565138]
[-0.67569672 -0.31302309]
[-0.82889394  1.65452915]
[-0.90665717  0.16570177]
[ 0.10949337  0.74834329]
[ 3.44000426  0.46549566]
[ 1.31016015 -0.78938156]
[-0.06711283 -0.2916635 ]
[-0.3452848  -0.49887447]
[ 2.90538524 -0.54957458]
[-0.02129948 -1.04210998]
[ 0.24549802  1.28736714]
[ 0.48050048  0.78520637]
[ 0.1798732   0.27700225]
[ 0.54007392  1.1493092 ]
[ 0.49128108  0.73589911]
[ 0.70527908  0.94299521]
[ 0.5551104   0.80324392]
[ 0.10838977  1.00422727]
[ 0.81401791  0.49429169]
[ 0.13558425  0.57465701]
[ 0.64571629  1.05884209]
[ 0.28760187  1.22394365]
[-0.16646451  1.02100742]
[ 0.04455893  0.97732399]
[ 0.4022937   0.93794506]
[ 0.38842358  1.1568409 ]
[ 0.49395257  0.59383325]
[ 0.61938344  0.5277949 ]
[ 0.13361651  0.57998565]
[ 0.57452375  0.51920849]
[ 0.91513906  0.68609191]
[ 0.59569512  0.5974255 ]
[ 0.21138503  1.35846046]
[ 0.8173645   0.77062769]
[ 0.74857432  0.8541002 ]
[ 0.58348391  0.781852  ]
[ 0.53987232  0.6351167 ]
[ 0.49382643  0.56534476]
[ 0.45257401  0.78289108]
[ 1.03743071  0.76905149]
[ 0.09227275  0.76175271]
[ 0.49590505  0.48132883]
[ 0.63254655  1.02165866]
[ 0.21629384  0.45043079]
```

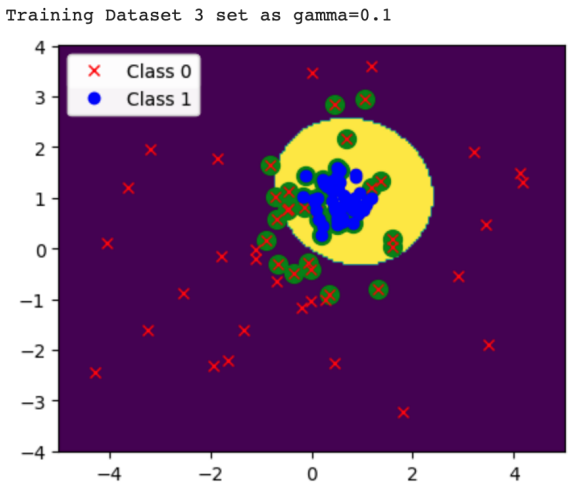
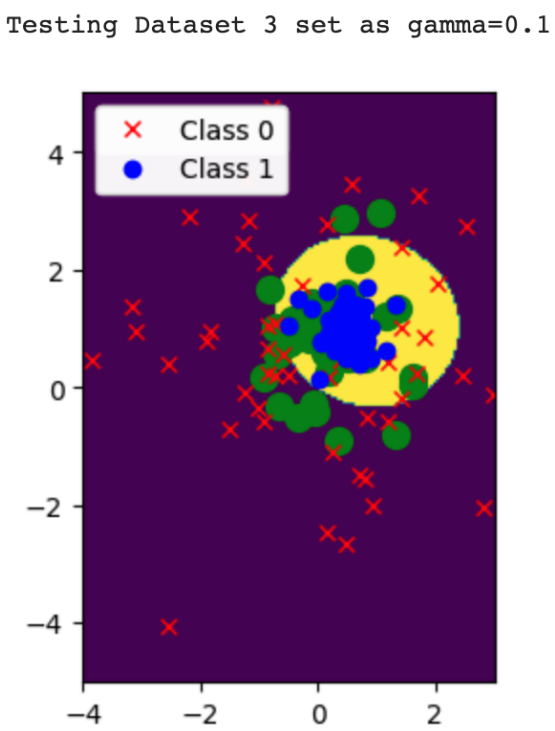
```
[-0.45506368  0.79536934]
[-0.44720807  1.13514243]
[ 3.20775231  1.91160833]
[ 1.35240975  1.32747597]
[-1.10754848 -0.19512252]
[-3.64007736  1.20889011]
[ 4.12858298  1.47898401]
[ 0.01816077  3.47565138]
[-0.67569672 -0.31302309]
[-0.82889394  1.65452915]
[-0.90665717  0.16570177]
[ 0.10949337  0.74834329]
[ 3.44000426  0.46549566]
[ 1.31016015 -0.78938156]
[-0.06711283 -0.2916635 ]
[-0.3452848  -0.49887447]
[ 2.90538524 -0.54957458]
[ 0.24549802  1.28736714]
[ 0.48050048  0.78520637]
[ 0.1798732   0.27700225]
[ 0.54007392  1.1493092 ]
[ 0.49128108  0.73589911]
[ 0.70527908  0.94299521]
[ 0.5551104   0.80324392]
[ 0.10838977  1.00422727]
[ 0.81401791  0.49429169]
[ 0.13558425  0.57465701]
[ 0.64571629  1.05884209]
[ 0.28760187  1.22394365]
[-0.16646451  1.02100742]
[ 0.04455893  0.97732399]
[ 0.4022937   0.93794506]
[ 0.38842358  1.1568409 ]
[ 0.49395257  0.59383325]
[ 0.61938344  0.5277949 ]
[ 0.13361651  0.57998565]
[ 0.57452375  0.51920849]
[ 0.91513906  0.68609191]
[ 0.59569512  0.5974255 ]
[ 0.8173645   0.77062769]
[ 0.74857432  0.8541002 ]
[ 0.58348391  0.781852  ]
[ 0.53987232  0.6351167 ]
[ 0.49382643  0.56534476]
[ 0.45257401  0.78289108]
[ 1.03743071  0.76905149]
[ 0.09227275  0.76175271]
[ 0.49590505  0.48132883]
[ 0.63254655  1.02165866]
[ 0.21629384  0.45043079]
[ 0.90797177  0.85680728]
[-0.11754044  1.44943913]
```

[0.06949687 0.91978049] [0.90797177 0.85680728] [-0.11754044 1.44943913] [0.36283865 1.20723961]]	[0.36283865 1.20723961]]

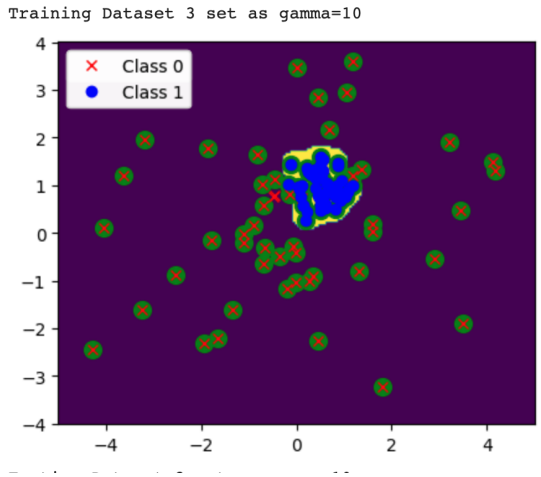
Use a Gaussian (RBF) Kernel with C parameter set to C = 1. Set γ = 0.1, 10, 200. Report the above items and also show the support vectors in the training-data plots for each value of γ. Explain the difference in decision regions for the different values of γ. Tip: you might want to try plots at other values of γ to help you understand its effects (no need to include these extra plots in your solution). Do you observe any overfitting or underfitting for any of the given values of γ? You will provide 6 plots in total.

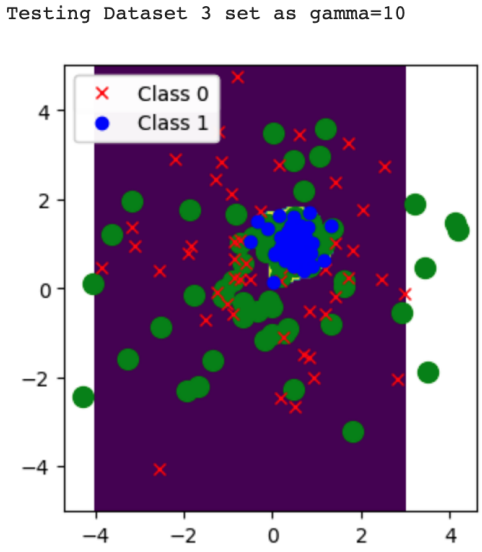
d) With different values of gamma, as the gamma value increases, the decision boundary becomes smaller so as an observation, as the gamma value increases we can see that it can result in overfitting. However, for small values of gamma, we can see that it could result to underfitting because of how big the decision boundary is for example when gamma is 0.1

When γ = 0.1

	
Training Accuracy is 89.0% when gamma=0.1 Testing Accuracy is 89.0% when gamma=0.1	

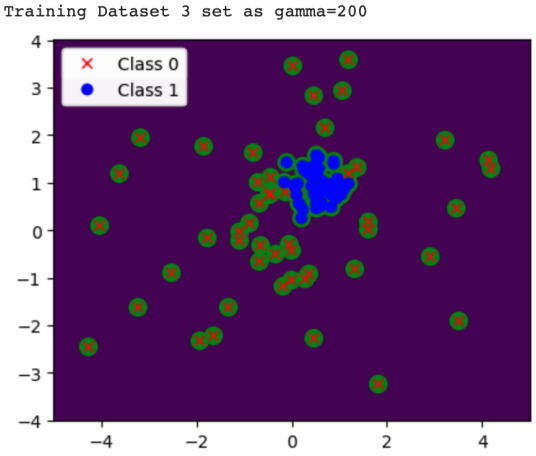
When γ = 10

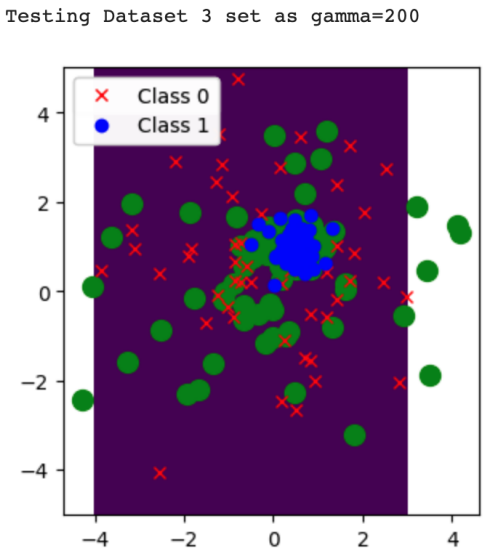




Training Accuracy is 98.0% when gamma=10
Testing Accuracy is 94.0% when gamma=10

When γ = 100



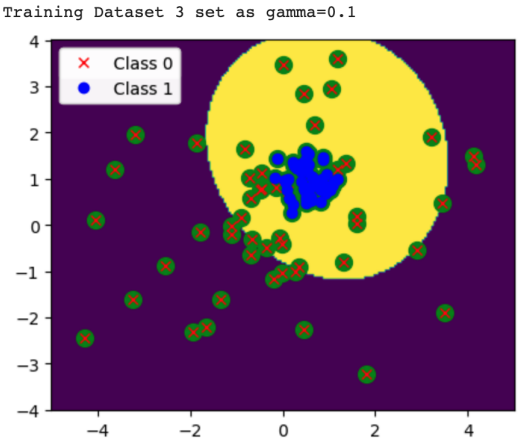
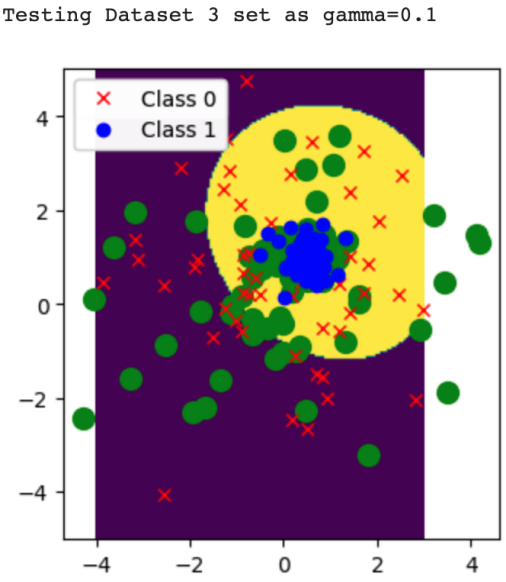


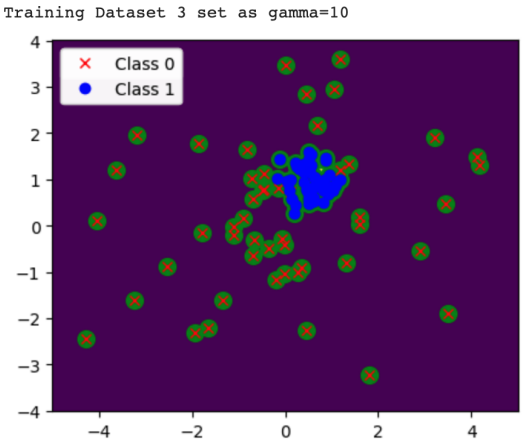
Training Accuracy is 98.0% when gamma=200
Testing Accuracy is 77.0% when gamma=200

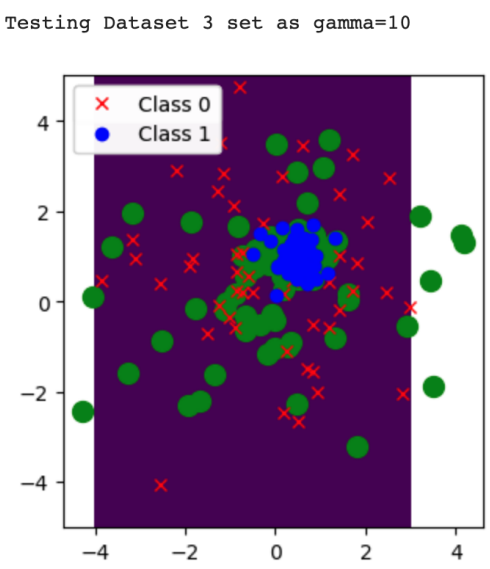
Use a Gaussian (RBF) Kernel and pick the γ parameter from part (d) (from the 3 given values) that results in the minimum test error. Set C = 0.01, 1, 100. Report the above items and also provide the support vectors in the plots for each value of C. Explain your observations in the different decision boundaries and the support vectors for the different values of C. You will provide 6 plots in total.

e) The gamma value that results in the minimum test error is when gamma is equal to 10. This was because when gamma is 10, the accuracies were improving and increasing. There was also no evidence of overfitting or underfitting and the support vectors were decreasing when C was increasing.

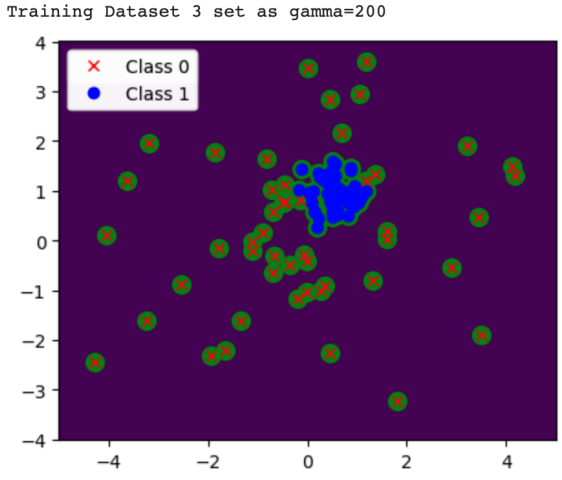
When C = 0.01

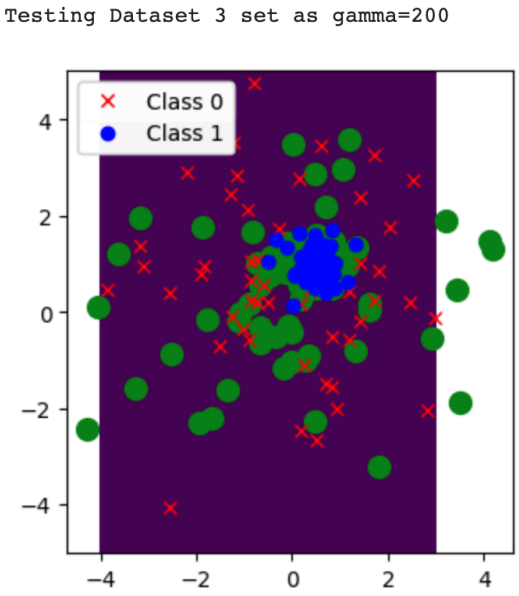
<table>
<tr>
<td> 
</td>
<td> 
</td>
</tr>
<tr>
<td>Training Accuracy is 73.0% when gamma=0.1
Testing Accuracy is 71.0% when gamma=0.1</td>
<td></td>
</tr>
</table>





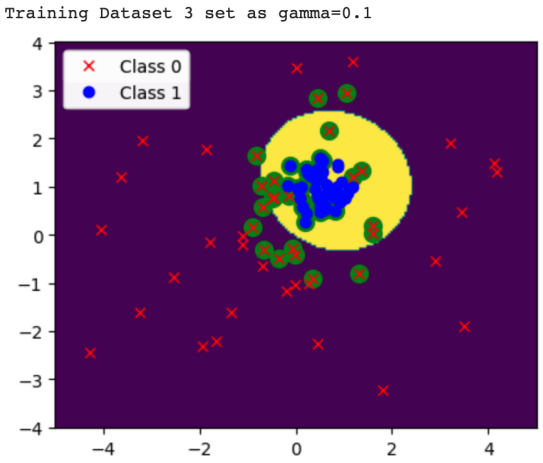
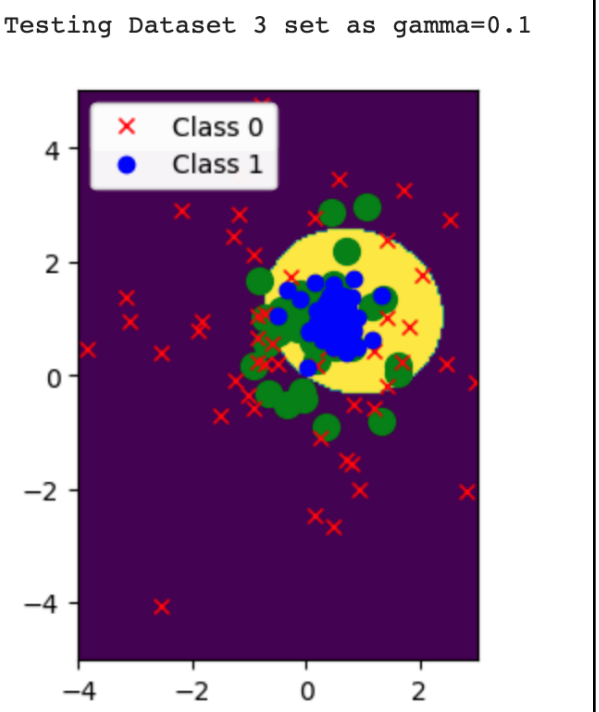
Training Accuracy is 88.0% when gamma=10
Testing Accuracy is 86.0% when gamma=10

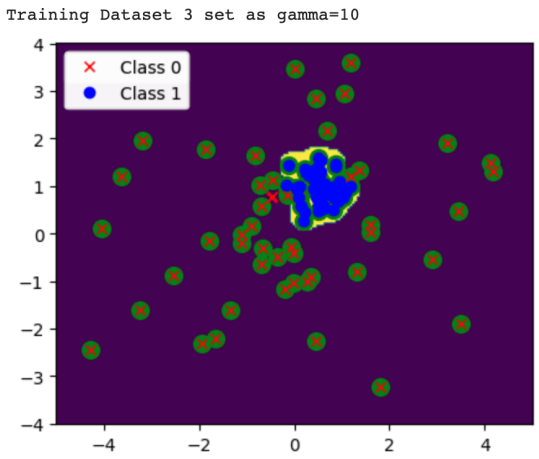
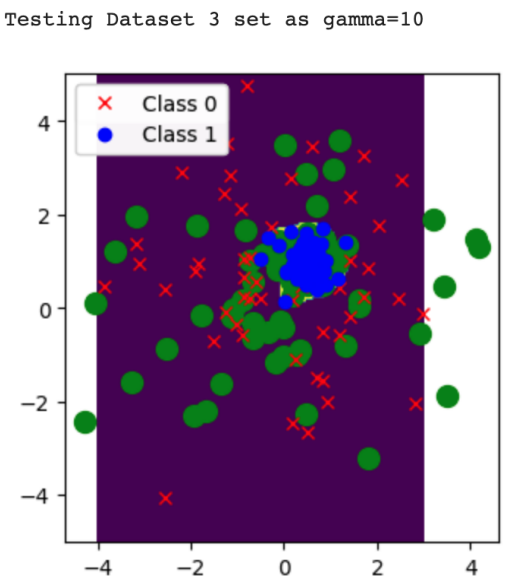


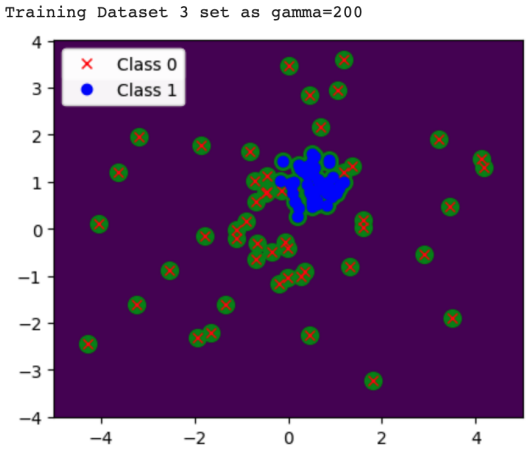
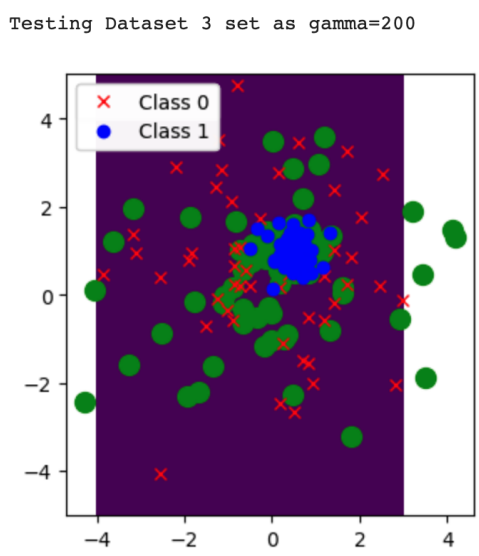


Training Accuracy is 99.0% when gamma=200
Testing Accuracy is 65.0% when gamma=200

When C = 1

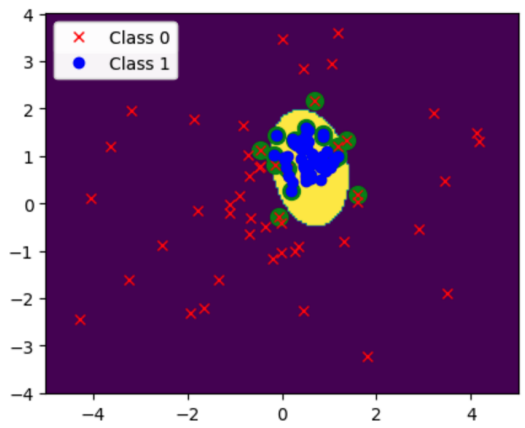
<table>
<tr>
<td> 
</td>
<td> 
</td>
</tr>
<tr>
<td>Training Accuracy is 89.0% when gamma=0.1
Testing Accuracy is 89.0% when gamma=0.1</td>
<td></td>
</tr>
</table>

	
Training Accuracy is 98.0% when gamma=10 Testing Accuracy is 94.0% when gamma=10	

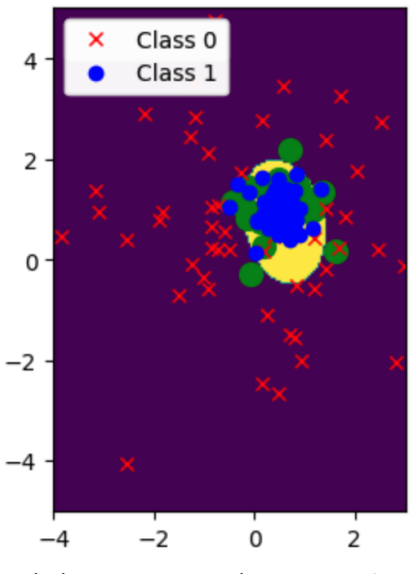
	
Training Accuracy is 98.0% when gamma=200 Testing Accuracy is 77.0% when gamma=200	

When C = 100

Training Dataset 3 set as gamma=0.1

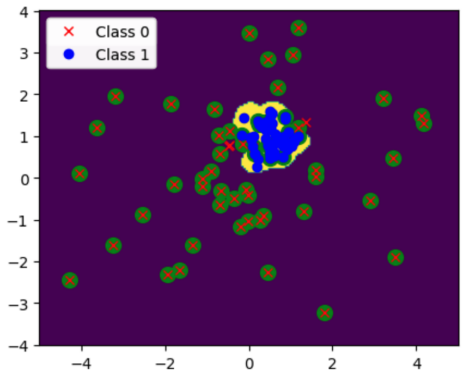


Testing Dataset 3 set as gamma=0.1

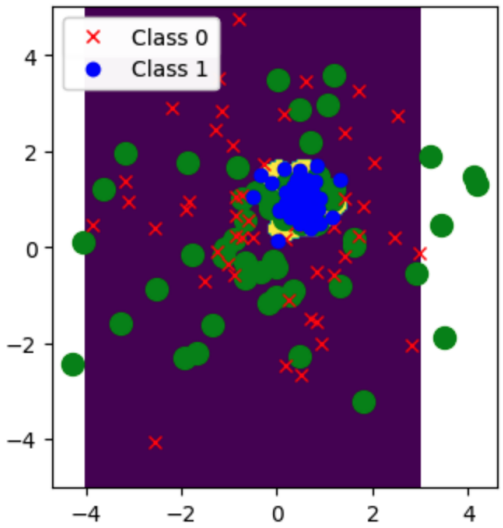


Training Accuracy is 96.0% when gamma=0.1
Testing Accuracy is 93.0% when gamma=0.1

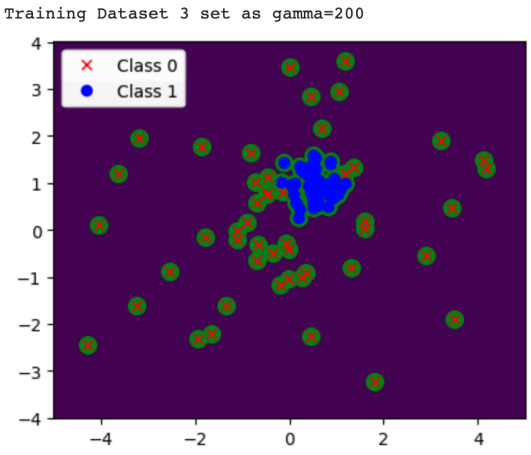
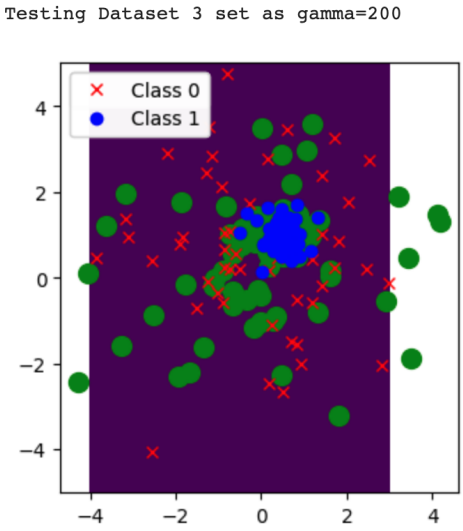
Training Dataset 3 set as gamma=10



Testing Dataset 3 set as gamma=10



Training Accuracy is 98.0% when gamma=10 Testing Accuracy is 95.0% when gamma=10	

	
Training Accuracy is 100.0% when gamma=200 Testing Accuracy is 77.0% when gamma=200	

2.

a) $J = -\sum_{i=1}^{C} p_i \ln \hat{p}_i$

Sol: Dimension of each of the ff quantities

① $\frac{\partial \hat{p}}{\partial a}$

$\hat{p}$ has $C \times 1$ dimension
a has $C \times 1$ dimension

$\therefore \frac{\partial \hat{p}}{\partial a}$ will have a $C \times C$ dimension

② $\nabla_{\hat{p}} J$ will have a $C \times 1$ dimensions

③ $\delta^{(L)} = \frac{\partial \vec{p}}{\partial a} \nabla_{\hat{p}} J = C \times 1$ dimensions

b) Find $\frac{\partial \hat{p}}{\partial a}$

Sol:

① $\hat{p} = \text{softmax}(a) = \begin{bmatrix} e^{a_1} / \sum_{k=1}^{C} e^{a_k} \\ e^{a_2} / \sum_{k=1}^{C} e^{a_k} \\ \vdots \\ e^{a_C} / \sum_{k=1}^{C} e^{a_k} \end{bmatrix}$

② $\frac{\partial \vec{p}}{\partial a} = \begin{bmatrix} \frac{\partial \hat{p}_1}{\partial a_1} & \frac{\partial \hat{p}_1}{\partial a_2} & \frac{\partial \hat{p}_1}{\partial a_3} & \cdots & \frac{\partial \hat{p}_1}{\partial a_C} \\ \frac{\partial \hat{p}_2}{\partial a_1} & & & & \\ \frac{\partial \hat{p}_3}{\partial a_1} & & & & \\ \vdots & & & & \\ \frac{\partial \hat{p}_C}{\partial a_1} & & & & \frac{\partial \hat{p}_C}{\partial a_C} \end{bmatrix}$

$\left[\frac{e^{a_i}}{\sum_{k=1}^{C} e^{a_k}} \right]^2$

③ $\frac{\partial \hat{p}_i}{\partial a_i} = \frac{e^{a_i}\left(\sum_{k=1}^{C} e^{a_k}\right) - e^{a_i}\left(e^{a_i}\right)}{\left(\sum_{k=1}^{C} e^{a_k}\right)^2} = \frac{e^{a_i}\left(\sum_{k=1}^{C} e^{a_k}\right)}{\left(\sum_{k=1}^{C} e^{a_k}\right)^2} - \frac{\left(e^{a_i}\right)\left(e^{a_i}\right)}{\left(\sum_{k=1}^{C} e^{a_k}\right)^2} = \frac{e^{a_i}}{\left(\sum_{k=1}^{C} e^{a_k}\right)} - \frac{\left(e^{a_i}\right)^2}{\left(\sum_{k=1}^{C} e^{a_k}\right)^2} = \hat{p}_i - (\hat{p}_i)^2$

$= \hat{p}_i (1 - \hat{p}_i)$

④ $\frac{\partial \hat{P}_i}{\partial a_j} = \frac{0 - e^{a_i}(e^{a_j})}{\left(\sum_{k=1}^{c} a^k\right)^2} = \frac{(-e^{a_i})(e^{a_j})}{\left(\sum_{k=1}^{c} a^k\right)^2} = -\hat{P}_i \hat{P}_j$

⑤ $\frac{\partial \hat{P}}{\partial a} = \begin{bmatrix} \hat{P}_1(1-\hat{P}_1) & -\hat{P}_1\hat{P}_2 & -\hat{P}_1\hat{P}_3 & \cdots & -\hat{P}_1\hat{P}_c \\ -\hat{P}_2\hat{P}_1 & \hat{P}_2(1-\hat{P}_2) & -P_2P_3 & & -\hat{P}_2P_2 \\ \vdots & & \hat{P}_3(1-\hat{P}_3) & & \vdots \\ -\hat{P}_c(\hat{P}_1) & \cdots & \cdots & & \hat{P}_c(1-\hat{P}_c) \end{bmatrix}$

c.) $\nabla_{\hat{P}} J$

Sol:

① $\nabla_{\hat{P}} J = \begin{bmatrix} \partial J/\partial \hat{P}_1 \\ \partial J/\partial \hat{P}_2 \\ \partial J/\partial \hat{P}_3 \\ \vdots \\ \partial J/\partial \hat{P}_c \end{bmatrix} = \begin{bmatrix} -P_1/\hat{P}_1 \\ -P_2/\hat{P}_2 \\ -P_3/\hat{P}_3 \\ \vdots \\ -P_c/\hat{P}_c \end{bmatrix}$

d) $s^{(L)} = \hat{P} - P$

Sol:

① $s^{(L)} = \frac{\partial \hat{P}}{\partial a} \cdot \nabla_{\hat{P}} J = \begin{bmatrix} \hat{P}_1(1-\hat{P}_1) & -\hat{P}_1\hat{P}_2 & \cdots & -\hat{P}_1\hat{P}_c \\ -\hat{P}_1\hat{P}_2 & \hat{P}_2(1-\hat{P}_2) & & -\hat{P}_2\hat{P}_2 \\ \vdots & & & \\ -\hat{P}_c\hat{P}_1 & -\hat{P}_c\hat{P}_2 & \cdots & \hat{P}_c(1-\hat{P}_c) \end{bmatrix} \begin{bmatrix} -P_1/\hat{P}_1 \\ -P_2/\hat{P}_2 \\ -P_3/\hat{P}_3 \\ \vdots \\ -P_c/\hat{P}_c \end{bmatrix} = \begin{bmatrix} -P_1(1-\hat{P}_1) + \hat{P}_1P_2 + \hat{P}_1P_3 + \cdots \hat{P}_1P_c \\ P_1\hat{P}_2 + (-P_2(1-\hat{P}_2)) + \hat{P}_2P_3 + \cdots \hat{P}_2P_c \\ \vdots \\ P_1\hat{P}_c + P_2\hat{P}_c + \cdots -P_c(1-\hat{P}_c) \end{bmatrix}$

$= \begin{bmatrix} -P_1 + \hat{P}_1 \sum_{i=1}^{c} P_i \\ -P_2 + \hat{P}_2 \sum_{i=1}^{c} P_i \\ -P_3 + \hat{P}_3 \sum_{i=1}^{c} P_i \\ \vdots \\ -P_c + \hat{P}_c \sum_{i=1}^{c} P_i \end{bmatrix}$

② $\sum_{i=1}^{c} P_i = 1$ since P_i are the probabilities

③ $s^{(L)} = \begin{bmatrix} \hat{P}_1 - P_1 \\ \hat{P}_2 - P_2 \\ \hat{P}_3 - P_3 \\ \vdots \\ \hat{P}_c - P_c \end{bmatrix} = \begin{bmatrix} \hat{P}_1 \\ \hat{P}_2 \\ \hat{P}_3 \\ \vdots \\ \hat{P}_c \end{bmatrix} - \begin{bmatrix} P_1 \\ P_2 \\ P_3 \\ \vdots \\ \hat{P}_c \end{bmatrix} = \hat{P} - P$

3)

3.

a.) Feed forward Computation

Sol: Find $\underline{a}^{(\ell)}$ and $\underline{v}^{(\ell)}$

①

$$\underline{\underline{W}}^{(1)} = \begin{bmatrix} 1 & -2 & 1 \\ 3 & 4 & -2 \end{bmatrix}, \quad \underline{b}^{(1)} = \begin{bmatrix} 1 \\ -2 \end{bmatrix}$$

$$\underline{a}^{(1)} = \underline{\underline{W}}^{(1)} x + \underline{b}^{(1)} \quad \text{(input vector)}$$

$$\underline{a}^{(1)} = \begin{bmatrix} 1 & -2 & 1 \\ 3 & 4 & -2 \end{bmatrix} \begin{bmatrix} 1 \\ -1 \\ 1 \end{bmatrix} + \begin{bmatrix} 1 \\ -2 \end{bmatrix} = \begin{bmatrix} 4 \\ -3 \end{bmatrix} + \begin{bmatrix} 1 \\ -2 \end{bmatrix} = \begin{bmatrix} 5 \\ -5 \end{bmatrix}$$

$$\underline{v}^{(1)} = \text{Relu}(\underline{a}^{(1)}) = \text{Relu}\left(\begin{bmatrix} 5 \\ -5 \end{bmatrix}\right) = \begin{bmatrix} 5 \\ 0 \end{bmatrix} \rightarrow \dot{\underline{v}}^{(1)} = \begin{bmatrix} 1 \\ 0.5 \end{bmatrix}$$

$$\rightarrow \underline{a}^{(1)} = \begin{bmatrix} 5 \\ -5 \end{bmatrix}, \quad \underline{v}^{(1)} = \begin{bmatrix} 5 \\ -5 \end{bmatrix} = 5 \begin{bmatrix} 1 \\ -1 \end{bmatrix}, \quad \dot{\underline{v}}^{(1)} = \begin{bmatrix} 1 \\ 0.5 \end{bmatrix}$$

② $\underline{\underline{W}}^{(2)} = \begin{bmatrix} 1 & -2 \\ 3 & 4 \end{bmatrix}, \quad \underline{b}^{(2)} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$

$$\underline{a}^{(2)} = \underline{\underline{W}}^{(2)} \underline{v}^{(1)} + \underline{b}^{(2)} = \begin{bmatrix} 1 & -2 \\ 3 & 4 \end{bmatrix} \begin{bmatrix} 5 \\ 0 \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 5 \\ 15 \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 6 \\ 15 \end{bmatrix}$$

($\underline{v}^{(1)}$ from ①)

$$\underline{v}^{(2)} = \text{Relu}(\underline{a}^{(2)}) = \text{Relu}\left(\begin{bmatrix} 6 \\ 15 \end{bmatrix}\right), \quad \dot{\underline{v}}^{(2)} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

$$\rightarrow \underline{a}^{(2)} = \begin{bmatrix} 6 \\ 15 \end{bmatrix}, \quad \underline{v}^{(2)} = \begin{bmatrix} 6 \\ 15 \end{bmatrix}, \quad \dot{\underline{v}}^{(2)} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

③ $\underline{\underline{W}}^{(3)} = \begin{bmatrix} 2 & 2 \\ 3 & -3 \\ 2 & 1 \end{bmatrix}, \quad \underline{b}^{(3)} = \begin{bmatrix} 0 \\ -4 \\ -2 \end{bmatrix}$

$$\underline{a}^{(3)} = \underline{\underline{W}}^{(3)} \underline{v}^{(2)} + \underline{b}^{(3)} = \begin{bmatrix} 2 & 2 \\ 3 & -3 \\ 2 & 1 \end{bmatrix} \begin{bmatrix} 6 \\ 15 \end{bmatrix} + \begin{bmatrix} 0 \\ -4 \\ 2 \end{bmatrix} = \begin{bmatrix} 42 \\ -27 \\ 27 \end{bmatrix} + \begin{bmatrix} 0 \\ -4 \\ -2 \end{bmatrix} = \begin{bmatrix} 42 \\ -31 \\ 25 \end{bmatrix}$$

$$\underline{v}^{(3)} = \begin{bmatrix} e^{42} / e^{42} + e^{-31} + e^{25} \\ e^{-31} / e^{42} + e^{-31} + e^{25} \\ e^{25} / e^{42} + e^{-31} + e^{25} \end{bmatrix} \rightarrow \underline{v}^{(3)} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$

b) Back-propagation Computation

Sol: Determine $\underline{\delta}^{(1)}$, provide the updated weights and biases

① $p = [0\ 0\ 1]^T$, $\hat{p} = [1\ 0\ 0]^T$

$$\delta^{(3)} = \hat{p} - p = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix}$$

② $\delta^{(2)} = (\underline{W}^{(3)T}\ \delta^{(3)}) \odot \dot{v}^{(2)} = \left(\begin{bmatrix} 2 & 3 & 2 \\ 2 & -3 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix} \right) \odot \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$

$\rightarrow s^{(2)} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$

③ $\delta^{(1)} = (\underline{W}^{(2)T}\ \delta^{(2)}) \odot \dot{v}^{(1)} = \left(\begin{bmatrix} 1 & 3 \\ -2 & 4 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} \right) \odot \begin{bmatrix} 1 \\ 0.5 \end{bmatrix} = \begin{bmatrix} 3 \\ 4 \end{bmatrix} \odot \begin{bmatrix} 1 \\ 0.5 \end{bmatrix} = \begin{bmatrix} 3 \\ 2 \end{bmatrix}$

$$W^{(1)} = W^{(1)} - \eta\, \delta^{(1)} x = \begin{bmatrix} 1 & -2 & 1 \\ 3 & 4 & -2 \end{bmatrix} - 0.5 \begin{bmatrix} 3 \\ 2 \end{bmatrix} [1\ -1\ 1] = \begin{bmatrix} 1 & -2 & 1 \\ 3 & 4 & -2 \end{bmatrix} - \begin{bmatrix} 3 & -3 & 3 \\ 2 & -2 & 2 \end{bmatrix} \cdot \frac{1}{2}$$

$$\underline{W}^{(1)} = \begin{bmatrix} -0.5 & -0.5 & -0.5 \\ 2 & 5 & -1 \end{bmatrix}$$

$$b^{(1)} = b^{(1)} - \eta \delta^{(1)} = \begin{bmatrix} 1 \\ -2 \end{bmatrix} - 0.5 \begin{bmatrix} 3 \\ 2 \end{bmatrix} = \begin{bmatrix} 1 \\ -2 \end{bmatrix} - \begin{bmatrix} 1.5 \\ 1 \end{bmatrix} = \begin{bmatrix} -0.5 \\ -3 \end{bmatrix}$$

④ $W^{(2)} = W^{(2)} - \eta\, \delta^{(2)} v^{(1)T} = \begin{bmatrix} 1 & -2 \\ 3 & 4 \end{bmatrix} - 0.5 \begin{bmatrix} 0 \\ 1 \end{bmatrix} [5\ 0] = \begin{bmatrix} 1 & -2 \\ 3 & 4 \end{bmatrix} - \begin{bmatrix} 0 & 0 \\ 2.5 & 0 \end{bmatrix}$

$$= \begin{bmatrix} 1 & -2 \\ 0.5 & 4 \end{bmatrix}$$

$$b^{(2)} = b^{(2)} - \eta \delta^{(2)} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} - 0.5 \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} - \begin{bmatrix} 0 \\ 1/2 \end{bmatrix} = \begin{bmatrix} 1 \\ -0.5 \end{bmatrix}$$

⑤ $W^{(3)} = W^{(3)} - \eta\, \delta^{(3)} v^{(2)T} = \begin{bmatrix} 2 & 2 \\ 3 & -3 \\ 2 & 1 \end{bmatrix} - 0.5 \begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix} [6\ 15] = \begin{bmatrix} 2 & 2 \\ 3 & -3 \\ 2 & 1 \end{bmatrix} - 0.5 \begin{bmatrix} 6 & 15 \\ 0 & 0 \\ -6 & -15 \end{bmatrix}$

$$= \begin{bmatrix} 2 & 2 \\ 3 & -3 \\ 2 & 1 \end{bmatrix} - \begin{bmatrix} 3 & 7.5 \\ 0 & 0 \\ -3 & -7.5 \end{bmatrix} = \begin{bmatrix} -1 & -5.5 \\ 3 & -3 \\ 5 & 8.5 \end{bmatrix}$$

$$b^{(3)} = b^{(3)} - \eta\, \delta^{(3)} = \begin{bmatrix} 0 \\ -4 \\ -2 \end{bmatrix} - 0.5 \begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix} = \begin{bmatrix} 0 \\ -4 \\ -2 \end{bmatrix} - \begin{bmatrix} 0.5 \\ 0 \\ -0.5 \end{bmatrix} = \begin{bmatrix} -0.5 \\ -4 \\ -1.5 \end{bmatrix}$$

c)

Sol:

① $a_1^{(3)} = 42$, $a_2^{(3)} = -31$, $a_3^{(3)} = 25$ → From part a

② $v_1^{(2)} = v_1^{(1)} + 1$, $v_2^{(2)} = 3v_1^{(1)}$, $v_1^{(1)} = x_1 - 2x_2 + x_3 + 1$, $v_2^{(1)} = 0$

③ $a_1^{(3)} = 2v_1^{(2)} + 2v_2^{(2)} + 0$

$= 2(v_1^{(1)} + 1) + 2(3v_1^{(1)})$

$= 2(x_1 - 2x_2 + x_3 + 1 + 1) + 6(x_1 - 2x_2 + x_3 + 1)$

$= (2x_1 - 4x_2 + 2x_3 + 4) + 6x_1 - 12x_2 + 6x_3 + 6$

$= 8x_1 - 16x_2 + 8x_3 + 10$

④ $a_2^{(3)} = 3v_1^{(2)} - 3v_2^{(2)} - 4$

$= 3(v_1^{(1)} + 1) - 3(3v_1^{(1)}) - 4$

$= 3(x_1 - 2x_2 + x_3 + 2) - 3[3(x_1 - 2x_2 + x_3 + 1)] - 4$

$= 3x_1 - 6x_2 + 3x_3 + 6 - 9x_1 + 18x_2 - 9x_3 - 9 - 4$

$= -6x_1 + 12x_2 - 6x_3 - 7$

⑤ $a_3^{(3)} = 2v_1^{(2)} + v_2^{(2)} - 2$

$= 2(v_1^{(1)} + 1) + 3v_1^{(1)} - 2$

$= 5v_1^{(1)} + 2 - 2$

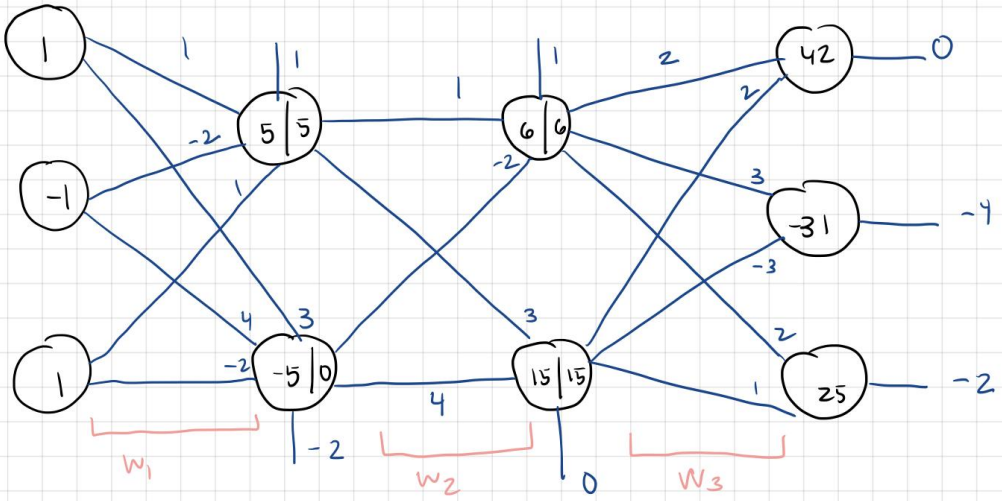
$= 5[x_1 - 2x_2 + x_3 + 1] = 5x_1 - 10x_2 + 5x_3 + 5$

⑥ $\begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} 8 & -16 & 8 \\ -6 & 12 & -6 \\ 5 & -10 & 5 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} 10 \\ -7 \\ 5 \end{bmatrix} = \begin{bmatrix} 8 & -16 & 8 \\ -6 & 12 & -6 \\ 5 & -10 & 5 \end{bmatrix} \begin{bmatrix} 1 \\ -1 \\ 1 \end{bmatrix} + \begin{bmatrix} 10 \\ -7 \\ 5 \end{bmatrix}$

$= \begin{bmatrix} 32 \\ -24 \\ 20 \end{bmatrix} + \begin{bmatrix} 10 \\ -7 \\ 5 \end{bmatrix} = \begin{bmatrix} 42 \\ -31 \\ 25 \end{bmatrix}$

⑦ $[\underline{W}_{eff}]x + \underline{b}_{eff} = \underline{a}^{(3)}$

$\underline{W}_{eff} = \begin{bmatrix} 8 & -16 & 8 \\ -6 & 12 & -6 \\ 5 & -10 & 5 \end{bmatrix}$, $b_{eff} = \begin{bmatrix} 10 \\ -7 \\ 5 \end{bmatrix}$



(Y) These effective weights and biases are a function of $\underline{x}$. Each layer of the artificial neural networks can be interpreted as a linear discriminant function. The ReLU activation is a non-linear activation function that introduces non-linearity into the neural network model. If a node of in the ANN is negative, the ReLU activation function sets its output to zero which removes the node from the networks computation. We can get the output values of the values of the final layer by the values of $\underline{W}_{eff}$ and $\underline{b}_{eff}$. The output values can be interpreted as the distance of each input row from the decision boundary of the classifier while every row of $\underline{W}_{eff}$ corresponds to the weights of n class