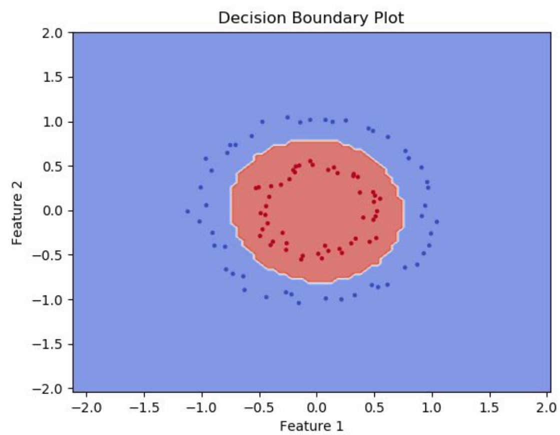
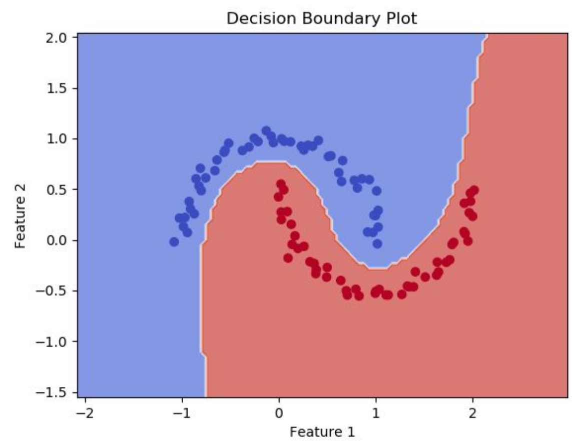
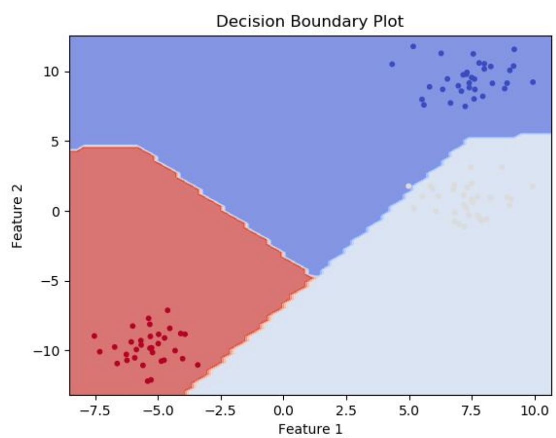
**ML Assignment 2 Report**
Avi Garg, 2017223

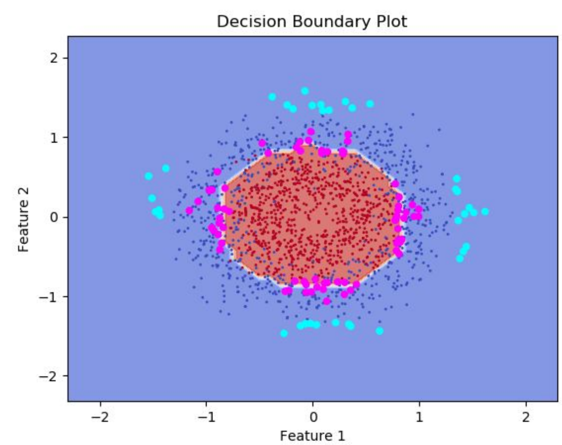
**Question1.**
Part a) Plots for all the datasets were made with their decision boundaries. For dataset 1, 2, 4, 5 the SVM is not linearly separable directly hence we used RBF kernel for the same.

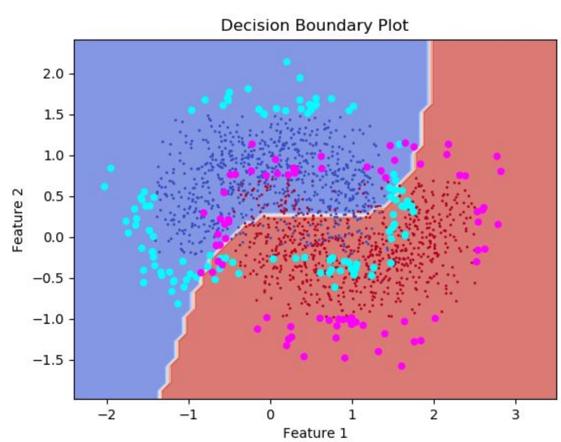
Part b) Respective plots were made for all 5 datasets using custom RBF kernel and for the 3rd dataset I used 1 vs all approach which was made from scratch with the linear kernel as well.



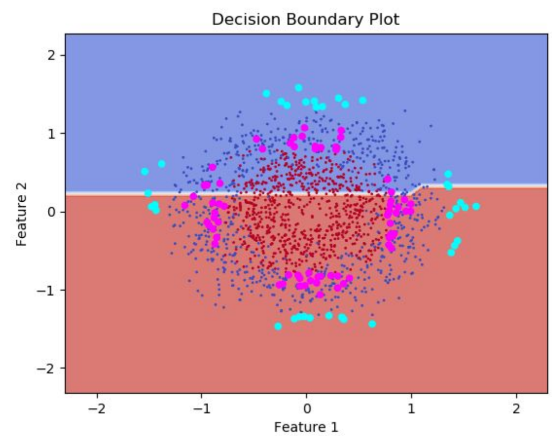



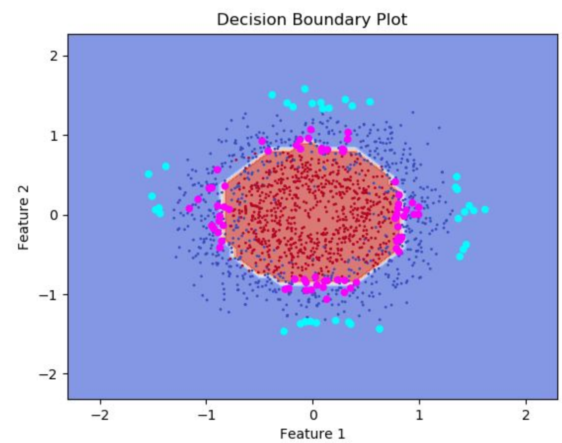



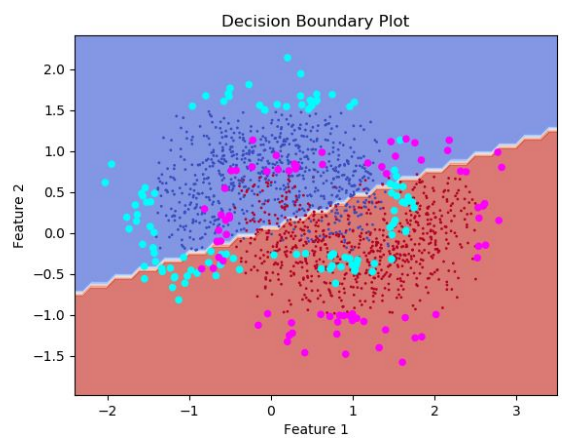



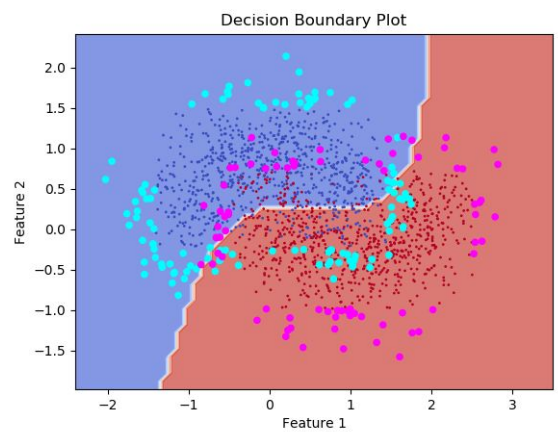
Part C) Outliers are marked separately with a bigger radius point and different shade, using the ZScore approach for individual classes. As shown in the below graphs for dataset 4 and 5.

Part D) Both kernels were used and clearly linear kernel performed poor as compared to RBF kernel, plots for which are shown below. Left graphs are for linear kernels.




The custom_predict function is performing equally good as the inbuilt predict function as can be seen for both the datasets in below outputs. For dataset 4 and 5 in order.

Accuracy for Prediciton with Custom Method

| | Linear | RBF |
|---|---|---|
| Train | 0.507 | 0.908 |
| Test | 0.515 | 0.91 |

Accuracy for Prediciton with Inbuilt Method

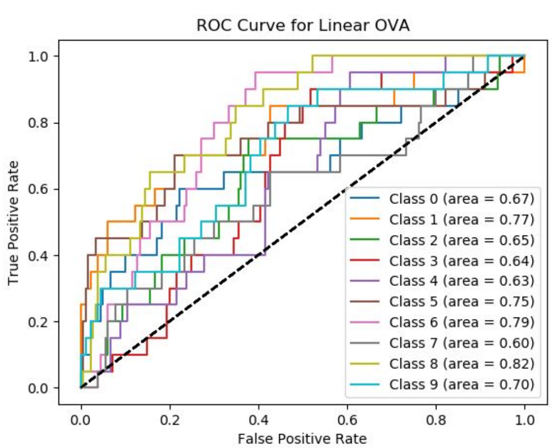
| | Linear | RBF |
|---|---|---|
| Train | 0.507 | 0.908 |
| Test | 0.515 | 0.91 |

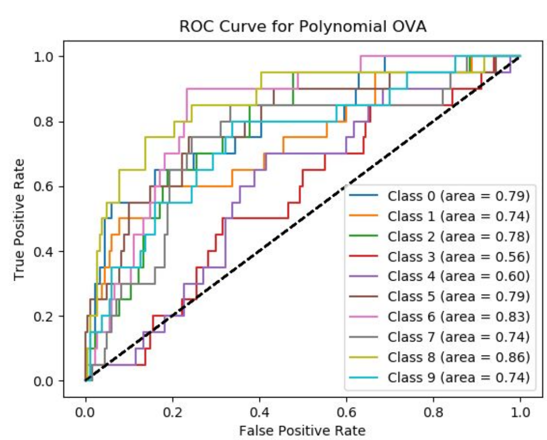
```
Accuracy for Prediciton with Custom Method
                  Linear                    RBF
  Train           0.885                    0.904
   Test            0.86                    0.876
Accuracy for Prediciton with Inbuilt Method
                  Linear                    RBF
  Train           0.885                    0.904
   Test            0.86                    0.876
```

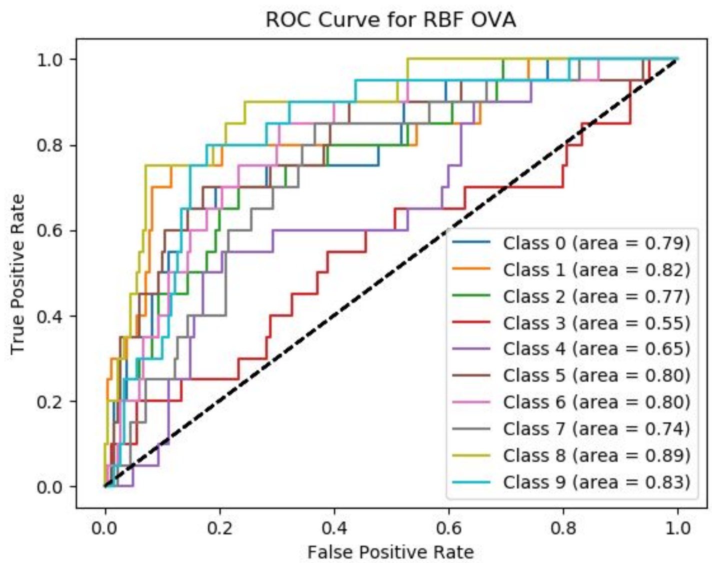
**Question2.**
RBF kernel performs the best as can be seen from the accuracies below for 5folds taking the maximum for each kernel.

```
Max accuracies for different kernels are:
Linear OVA: 0.27
RBF OVA: 0.36
Polynomial OVA: 0.325
Linear OVO: 0.295
RBF OVO: 0.33
Polynomial OVO: 0.32
It's clear that RBF is a better performing kernel
```

We plot the ROC curves to verify the same, also we have the confusion matrix, with accuracy on the validation set, with classification report which gave us these results. The ROC curves for some of the best models on basis of accuracy are below.






Confusion matrix and other visualizations are printed in the terminal but were not pasted here please refer to code.

**Question3.**

Q3. XOR is not linearly seperable, however, using Kernel, we can make it seperable.
Take transformation.

- $\phi\left(\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}\right) \rightarrow \begin{bmatrix} x_1 \\ x_2 \\ x_1 x_2 \end{bmatrix}$, we get.

| $x_1$ | $x_2$ | $x_1 \cdot x_2$ | $x_1 \oplus x_2$. |
|---|---|---|---|
| 0 | 0 | 0 | 1 |
| 0 | 1 | 0 | 0 |
| 1 | 0 | 0 | 0 |
| 1 | 1 | 1 | 1 |

- Which leads to the decision boundary $2x_1x_2 - x_1 - x_2 + 0.5$ and correctly classifies XOR operator.

Process:

$(x_1, x_2) = (0,0)$ $\quad 2x_1x_2 - x_1 - x_2 + 0.5 = 0.5 \in C_1$

$(x_1, x_2) = (0,1)$ $\quad$ " $= -0.5 \in C_0$.

$(x_1, x_2) = (1,0)$ $\quad$ " $= -0.5 \in C_0$.

$(x_1, x_2) = (1,1)$ $\quad$ " $= 0.5 \in C_1$

$\therefore$ XOR is modelled using as SVM, correctly & taking $y = 2x_1x_2 - x_1 - x_2 + 1$, one gets exact value of $y$ as required.



We can model a XOR using kernel techniques.

**Question4.** 6 features are there.

Q4. $K(x, x') = (1 + x^T x')^2$ , $x = [x_1\ x_2]^T$

$\therefore K(x, x') = (1 + x_1 x_1' + x_2 x_2')^2$

$= 1 + (x_1)^2 (x_1')^2 + (x_2)^2 (x_2')^2 + (\sqrt{2} x_1)(\sqrt{2} x_1')$

$+ (\sqrt{2} x_2)(\sqrt{2} x_2')$

$+ (\sqrt{2} x_1 x_2)(\sqrt{2} x_1' x_2')$

$$= [1\ \ x_1^2\ \ x_2^2\ \ \sqrt{2} x_1\ \ \sqrt{2} x_2\ \ \sqrt{2} x_1 x_2]^T \begin{bmatrix} 1 \\ x_1'^2 \\ x_2'^2 \\ \sqrt{2} x_1' \\ \sqrt{2} x_2' \\ \sqrt{2} x_1' x_2' \end{bmatrix}$$

$\therefore K(x, x') = \phi(x)^T \phi(x')$

$\therefore \phi(x) = [1\ \ x_1^2\ \ x_2^2\ \ \sqrt{2} x_1\ \ \sqrt{2} x_2\ \ \sqrt{2} x_1 x_2]$

$\therefore w = \sum_{i=1}^{N} a_i d_i \phi(x_i)$. $\phi(x)$ has 6 features.

**Question5.**

5.) $w = \sum \alpha_i y^{(i)} x^{(i)}$, substituting the same,

we have $w = (2.008, 3.8712)$

For $b$, averaging on values of $y^{(i)} = +1, -1$.

$$\therefore b = -\frac{1}{2}\left(\min_{i:\, y^{(i)}=+1} w^T x^{(i)} + \max_{i:\, y^{(i)}=-1} w^T x^{(i)}\right).$$

$$\therefore b = -\frac{1}{2}(16.186 + 12.14552) = -14.16576.$$

$\therefore$ Decision boundary $\rightarrow$ $w^T x + b =$

$2.008 x_1 + 3.8712 x_2 - 14.11576$.

For $\alpha = 1.18$ i.e. for $(2.5, 1)$ & $(3.5, 4)$ is large they must be outliers.

Thus support vectors corresponding to $\alpha \neq 0$ are.

$(4, 2.9)$ & $(2, 2.1)$.

For $(3,3)$, $w^T x + b = 5.47184 \geq 0$ $\therefore (3,3)$ lies in class 1.

**Question6.**

6.) $K(u, v) = e^{-\gamma \|u - v\|^2}$

For $u, v$ real numbers, we have,

$K(u, v) = e^{-\gamma (u-v)^2}$ ; for simplicity take $\gamma = 1$

$\therefore K(u, v) = e^{-u^2} e^{-v^2} e^{2uv}$.

Using taylor series, we have,

$$k(u,v) = e^{-u^2} \cdot e^{-v^2} \sum_{k=1}^{\infty} \frac{2^k u^k v^k}{k!}$$

$$= \sum_{k=1}^{\infty} \frac{2^k \left(u^k e^{-u^2}\right)\left(v^k e^{-v^2}\right)}{k!}.$$

$$= \phi(u)^T \phi(v).$$

Now $\phi(u) = \left[ e^{-u^2}, \sqrt{\frac{2}{1!}}\, u e^{-u^2}, \ldots \sqrt{\frac{2^k}{k!}}\, u^k e^{-u^2}, \ldots \right]^T$



Thus, $\phi(u)$ becomes an $\infty$ dimensional feature vector,

Also for high order terms say coeff. of $u^{10^6}$ will be $\sqrt{\frac{2^{10^6}}{10^6!}}\, e^{-u^2}$, will be small as $2^{10^6} \ll (10^6!)$

And as $k \to \infty$, $\lim_{k \to \infty} \frac{2^k}{k!} \to 0$, i.e. will be small.