Assignment 3

Report file

Shivani Panchiwala

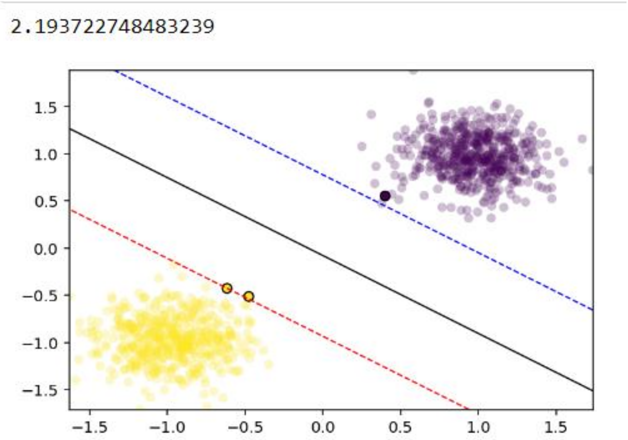
UTA ID: 1001982478

CSE – 6363-001

Prof. Alex Dhillof

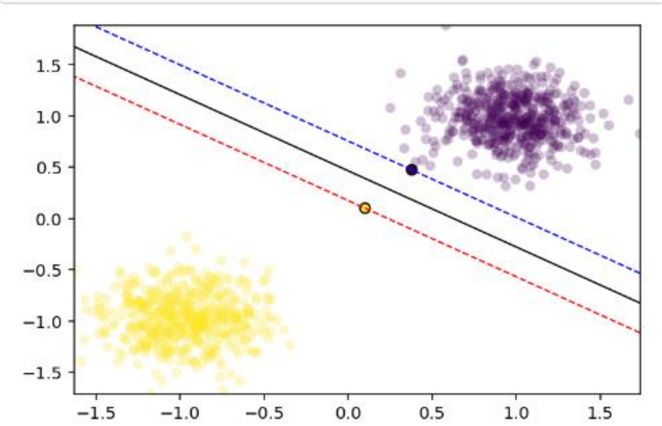
Using the linear kernel:

The circled points are our support vectors. They should be the only points with non-zero αα values.

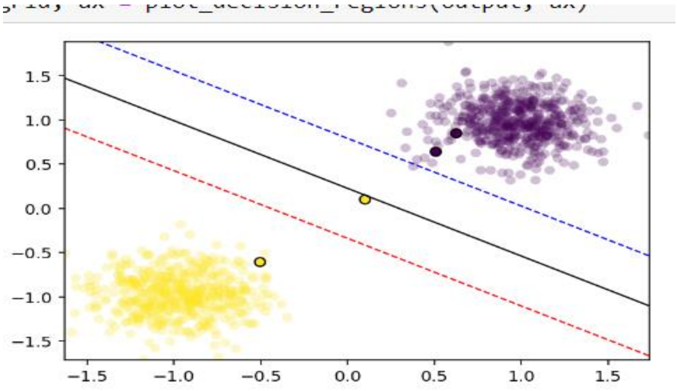


```
2.193722748483239
```

Model performs with an outlier.

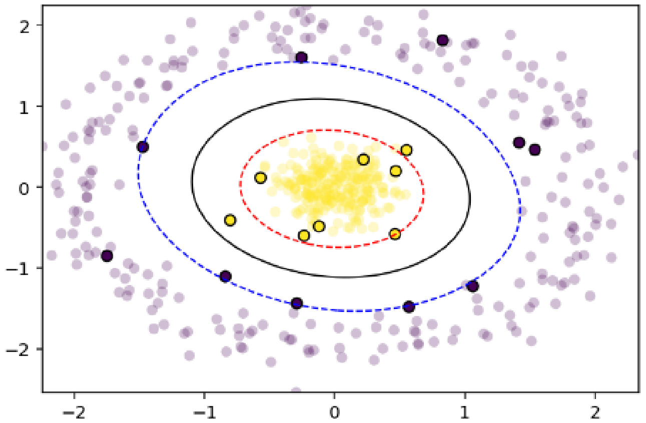


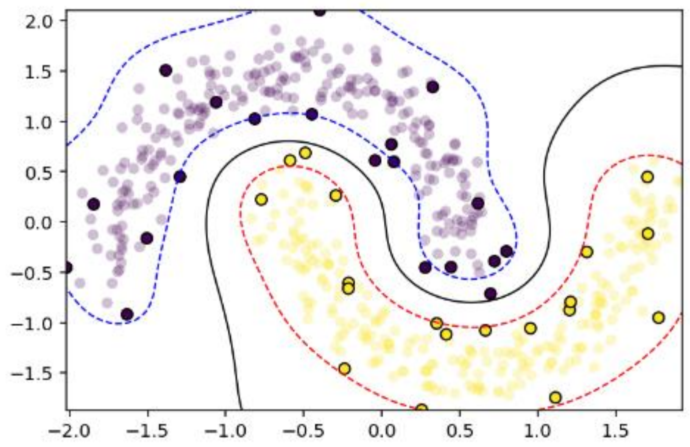
The outlier won't influence our decision region as much.



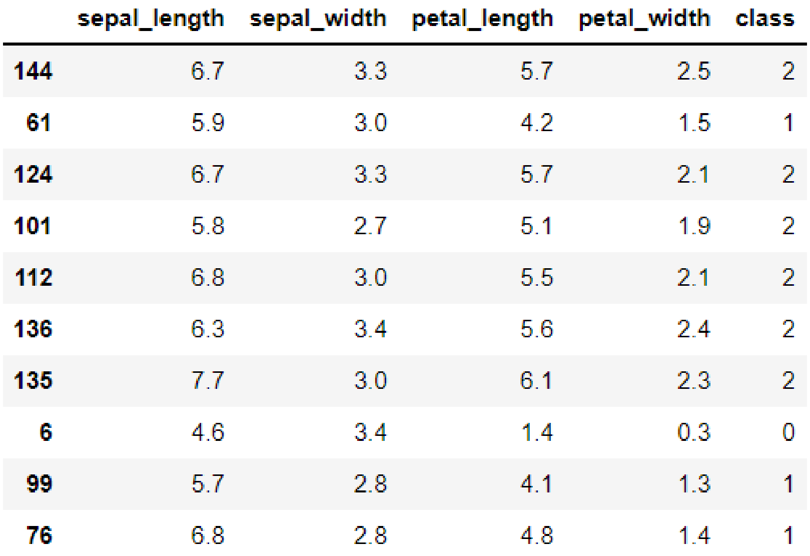
```
grid, ax = plot_decision_regions(output, ax)
```

Using Non linear Kernel:

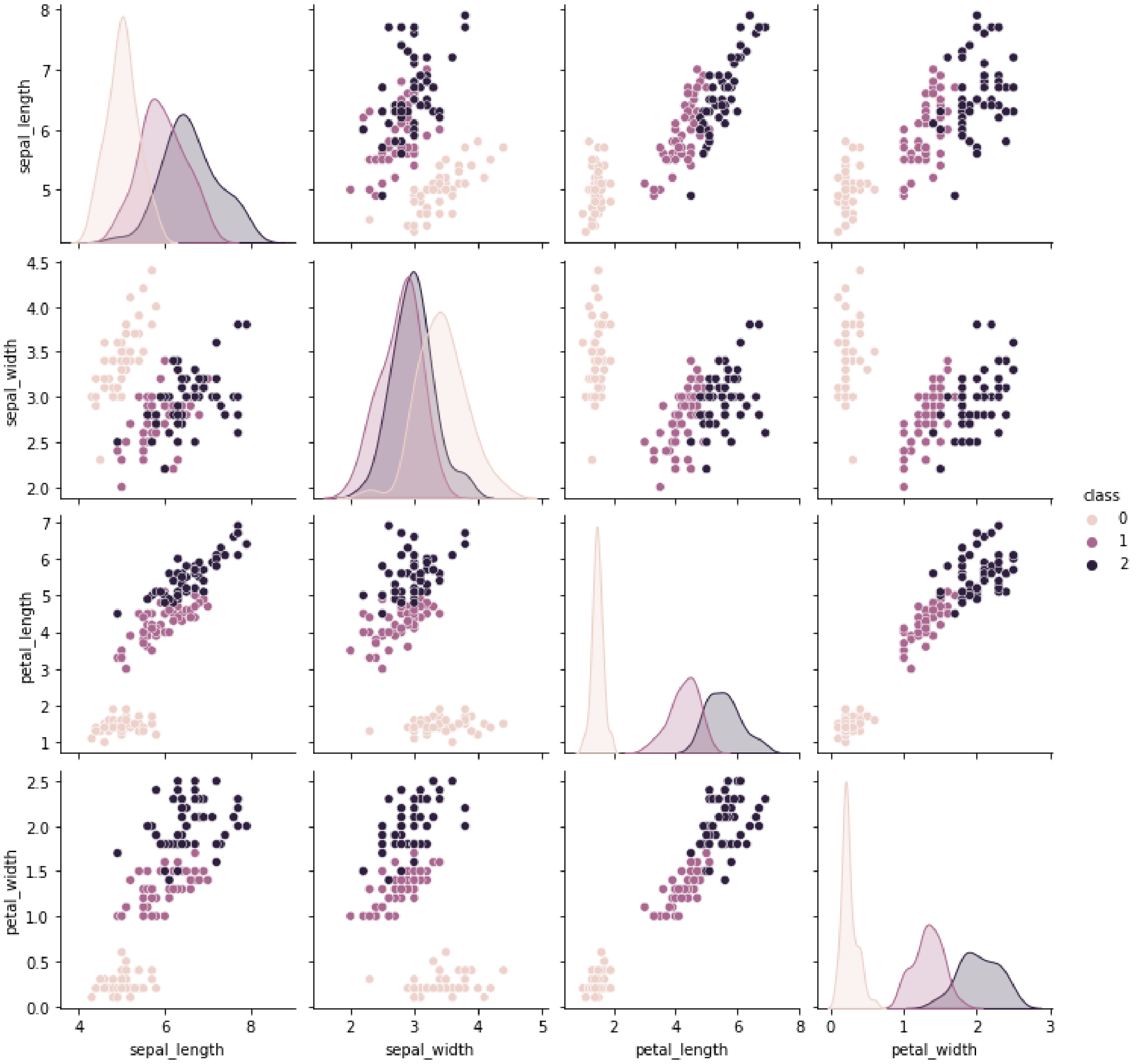




Multiclass _ SVM:

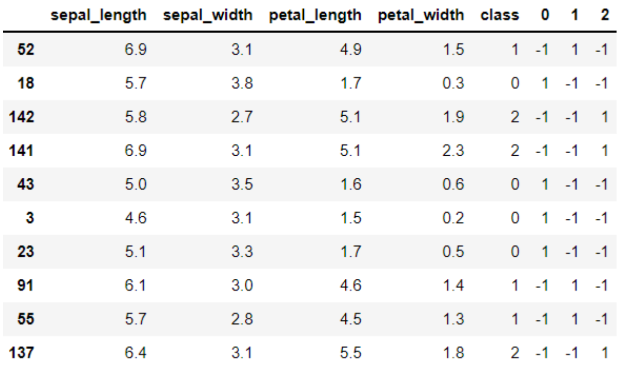


	sepal_length	sepal_width	petal_length	petal_width	class
144	6.7	3.3	5.7	2.5	2
61	5.9	3.0	4.2	1.5	1
124	6.7	3.3	5.7	2.1	2
101	5.8	2.7	5.1	1.9	2
112	6.8	3.0	5.5	2.1	2
136	6.3	3.4	5.6	2.4	2
135	7.7	3.0	6.1	2.3	2
6	4.6	3.4	1.4	0.3	0
99	5.7	2.8	4.1	1.3	1
76	6.8	2.8	4.8	1.4	1



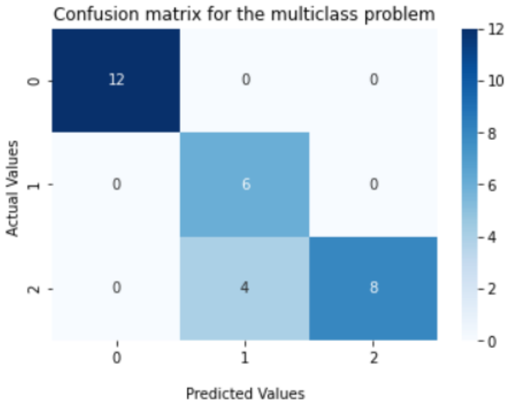
It can be clearly seen that flowers in class 0 can be linearly separated from other types in flowers.

Now for the *one vs all* method we need to train the model for data tagged as belonging to class i (+1) and data not associated with this class (-1).
When i = 0,1,2.

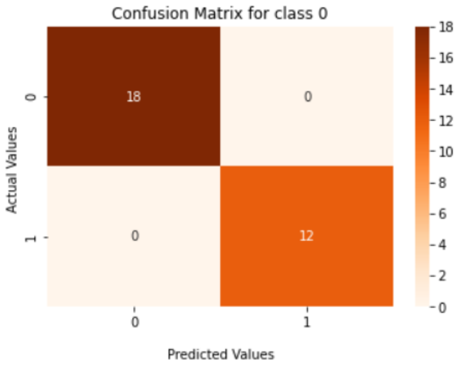


	sepal_length	sepal_width	petal_length	petal_width	class	0	1	2
52	6.9	3.1	4.9	1.5	1	-1	1	-1
18	5.7	3.8	1.7	0.3	0	1	-1	-1
142	5.8	2.7	5.1	1.9	2	-1	-1	1
141	6.9	3.1	5.1	2.3	2	-1	-1	1
43	5.0	3.5	1.6	0.6	0	1	-1	-1
3	4.6	3.1	1.5	0.2	0	1	-1	-1
23	5.1	3.3	1.7	0.5	0	1	-1	-1
91	6.1	3.0	4.6	1.4	1	-1	1	-1
55	5.7	2.8	4.5	1.3	1	-1	1	-1
137	6.4	3.1	5.5	1.8	2	-1	-1	1

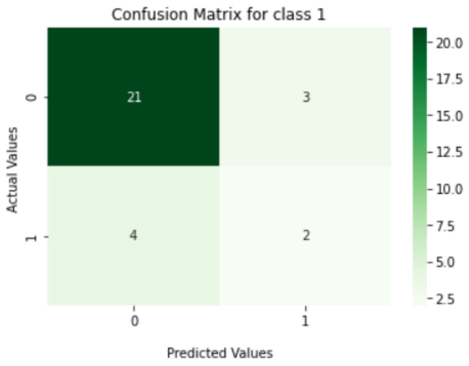
Confusion Matrix and Accuracy using Linear kernel:


Confusion matrix for the multiclass problem

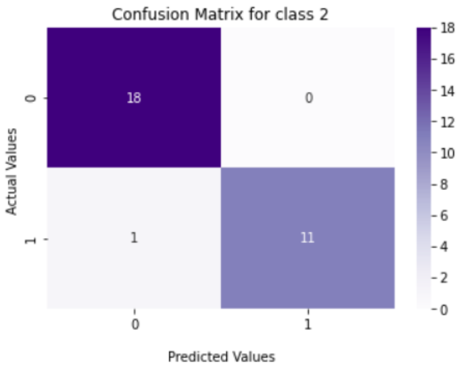
```
Accuracy: 86.67%
```


Confusion Matrix for class 0

```
Accuracy: 100.00%

Sensitivity: 100.00%
```

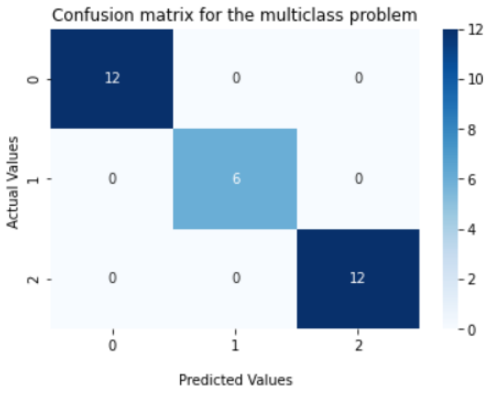

Confusion Matrix for class 1

```
Accuracy: 76.67%

Sensitivity: 33.33%
```

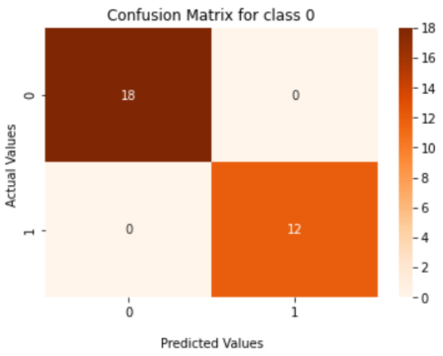

Confusion Matrix for class 2

```
Accuracy: 96.67%

Sensitivity: 91.67%
```

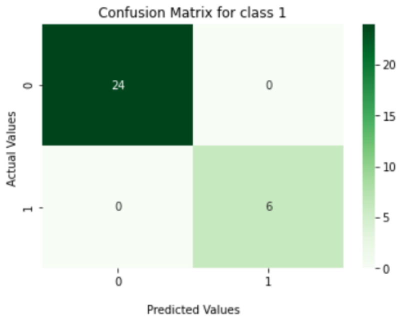
Confusion Matrix and Accuracy using Non Linear kernel:


Confusion matrix for the multiclass problem

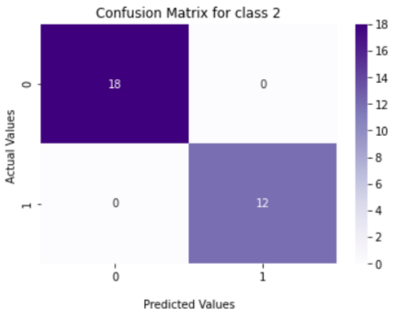
```
Accuracy: 100.00%
```


Confusion Matrix for class 0

```
Accuracy: 100.00%

Sensitivity: 100.00%
```



```
Accuracy: 100.00%

Sensitivity: 100.00%
```



```
Accuracy: 100.00%

Sensitivity: 100.00%
```

The Non-linear kernel seems to be doing well than linear kernel. Non linear kernel is more accurate than liner kernel.

References:

https://jonchar.net/notebooks/SVM/

https://github.com/itsikad/svm-smo/blob/main/example.py

https://github.com/itsikshteinberger/Sequential-Minimal-Optimization/blob/main/SMO.ipynb

https://stackoverflow.com/questions/28545318/svm-problems-with-smo-algorithm