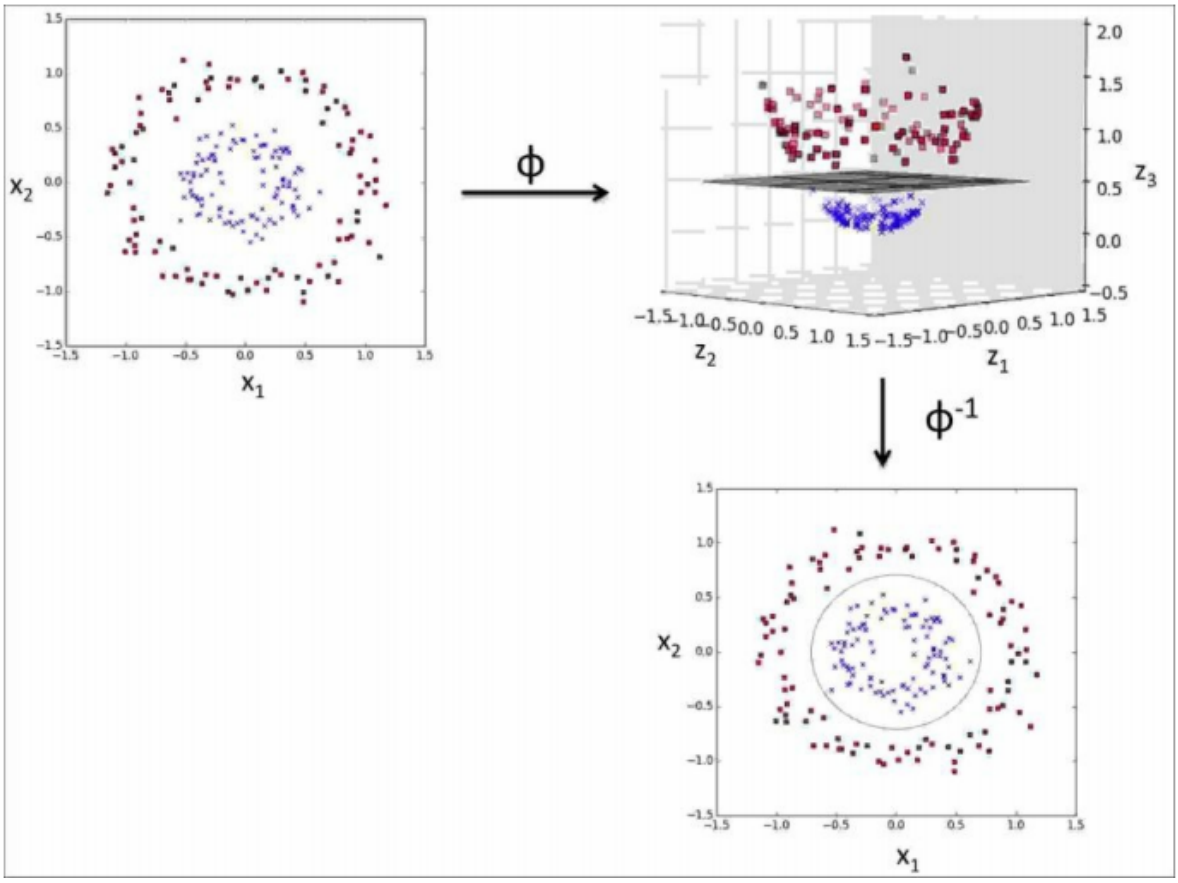
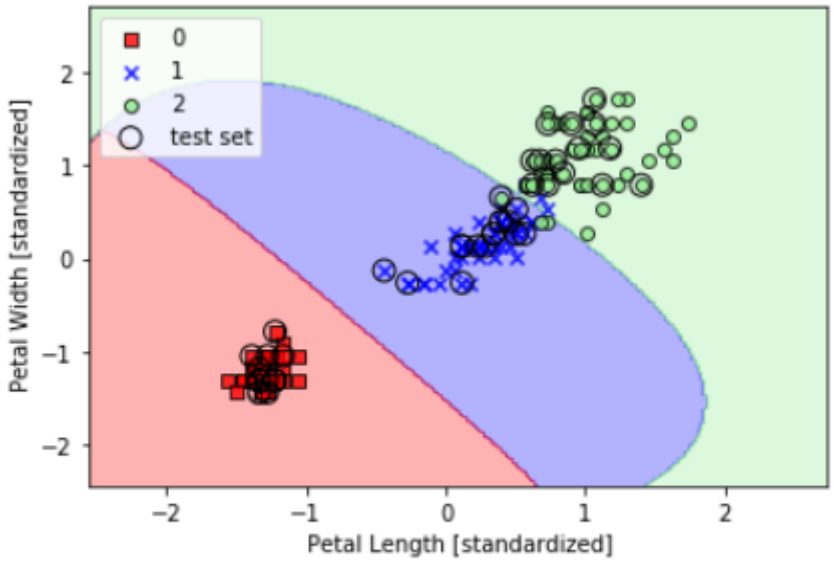
Kernel rbf

Kernel rbf is a method of classifying linearly non-separable features. The the rbf method allows features to become linearly separable by mapping features to a higher dimensional space.



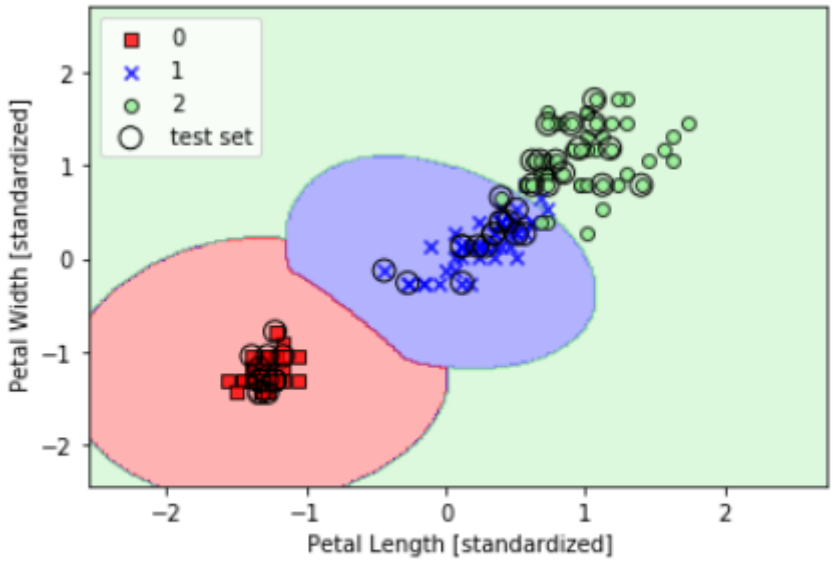

```
feature_names = [' Sepal Length', ' Sepal Width', ' Petal Length', ' Petal
Width']
X = iris.data[:, [2, 3]]
accuracy = train_smv_model(X=X,y=y,
                           feature_names=[' Petal Length', ' Petal Width'],
                           C=1.0, gamma=0.3, kernel='rbf', show_graph=True)

print('Accuracy: %.2f' % accuracy)
```




```
Accuracy: 0.98
```

The problem with this method is the gamma parameter. Increasing the value of gamma will increase the influence of the training sample. For example the image above has a gamma of 0.3.




```
Accuracy: 0.98
```

Now, while the image above has a 0.98 accuracy, Its bad for generalization, which is what we want in machine learning.

	kernel	accuracy	gamma	C_regularization parameter
0	rbf	0.9777777777777777	0.1	1.0
1	rbf	0.9555555555555556	0.2	1.0
2	rbf	0.9777777777777777	0.3	1.0
3	rbf	0.9777777777777777	0.4	1.0

Running the simulation 4 times slightly increasing the gamma parameter. Doesn't change the accuracy much, however; the higher the gamma parameter the worst the generalization.

Kernel Linear

Why use linear over the other kernels? Linear is usually the simplest. I would always try the linear method first, if the accuracy is suffice, then its good enough. The reason for this is that nonlinear kernels like rbf requires far more resources to train. Its very expensive to map features into higher dimensions especially when there's a lot of features and a whole lot of data. Looking at the table, linear seems to be the way to go, with an accuracy of 97% for all gamma parameters.

	kernel	accuracy	gamma	C_regularization parameter
0	linear	0.9777777777777777	0.1	1.0
1	linear	0.9777777777777777	0.2	1.0
2	linear	0.9777777777777777	0.3	1.0
3	linear	0.9777777777777777	0.4	1.0

Kernel Polynomial

	kernel	accuracy	gamma	C_regularization parameter
0	poly	0.7555555555555555	0.1	1.0
1	poly	0.9111111111111111	0.2	1.0
2	poly	0.8888888888888888	0.3	1.0
3	poly	0.9555555555555556	0.4	1.0

Looking at the table it seems linear has it beat, but comparing it to rbf. The rbf wins in accuracy on all gamma parameters. Polynomial and linear is usually less accurate, but are less time consuming.

Kernel Sigmoid

Sigmoid Kernel: K(X,Y)=tanh(γ·XTY+r)K(X,Y)=tanh(γ·XTY+r) , which is similar to the sigmoid function for the logistic regression in the previous assignment. However, in my previous assignment logistic regression model performed much better on all features, then what was shown here.

	kernel	accuracy	gamma	C_regularization parameter
0	sigmoid	0.9777777777777777	0.1	1.0
1	sigmoid	0.9111111111111111	0.2	1.0
2	sigmoid	0.8444444444444444	0.3	1.0
3	sigmoid	0.8	0.4	1.0

Looking at the table, Sigmoid seems to have performed the worst.

Conclusion

Based on the data and features used, the best classification method would be the linear kernel. Even if the other kernel methods did not perform poorly. The linear method gives the highest accuracy on all gamma parameters. In addition to the higher accuracy, linear is the least resource intensive method. In conclusion, In the case of the iris data set, use the linear method.