Implement the two non-linear kernels. You should be able to classify very hard data sets with these.

For polynomial kernels, the kernel function is , controls the complexity of model. When the model becomes more complex, it can represent the training set more accurately with higher variance, it may cause overfitting. While lower p could reduce the complexity, but may produce lower variance predictions when applied beyond the training set. In this kernel function, the linear shapes are not fitting to the data. With the increase of p, the decision boundary became wider. But when p > 3, with the increase of p, the decision boundary became thinner.

For polynomial kernels, the kernel function is $\kappa(\vec{x}, \vec{y}) = (\vec{x}^T \cdot \vec{y} + 1)^p$, p controls the complexity of model. When the model becomes more complex, it can represent the training set more accurately with higher variance, it may cause overfitting. While lower p could reduce the complexity, but may produce lower variance predictions when applied beyond the training set. In this kernel function, the linear shapes are not fitting to the data. With the increase of p, the decision boundary became wider. But when p > 3, with the increase of p, the decision boundary became thinner.

Explore the role of the slack parameter C. What happens for very large/small values?

C controls the cost of misclassification on the training data. Small C makes the cost of misclassificaiton low, more slack Large C makes the cost of misclassification high, smaller slack

The non-linear kernels have parameters; explore how they influence the decision boundary. Reason about this in terms of the bias variance trade-off.

Q5 Imagine that you are given data that is not easily separable. When should you opt for more slack rather than going for a more complex model and vice versa?¶

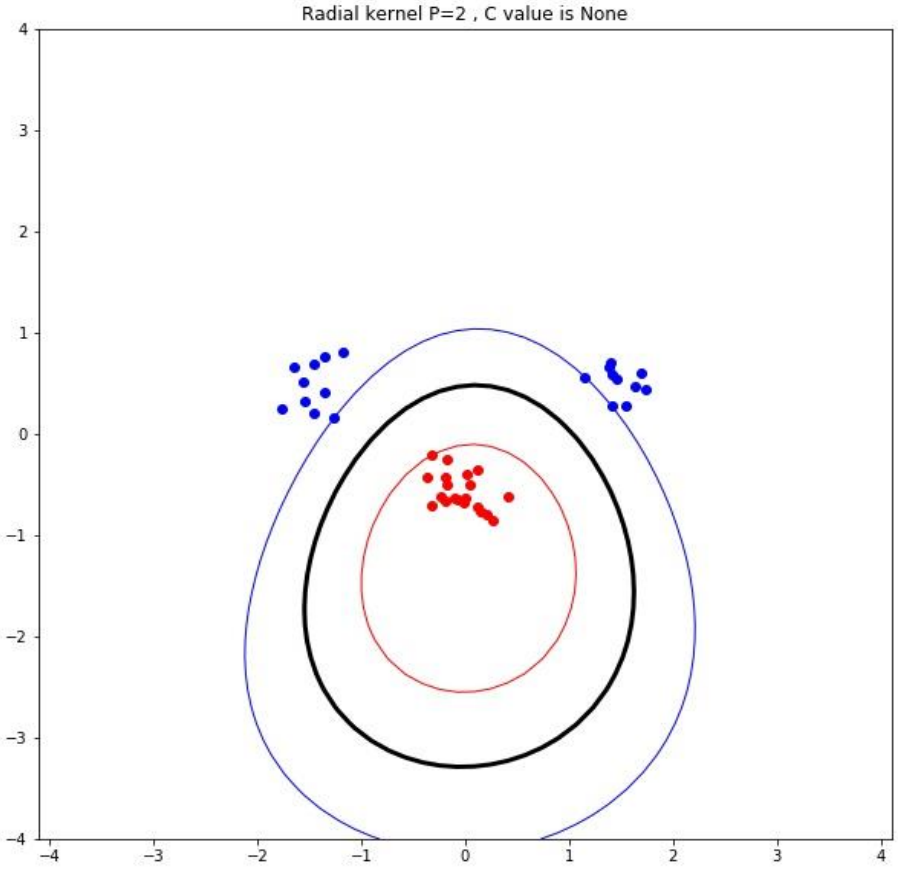
Large C gives a hypothesis of low bias high variance --> overfitting, more accurate Small C gives a hypothesis of high bias low variance --> underfitting

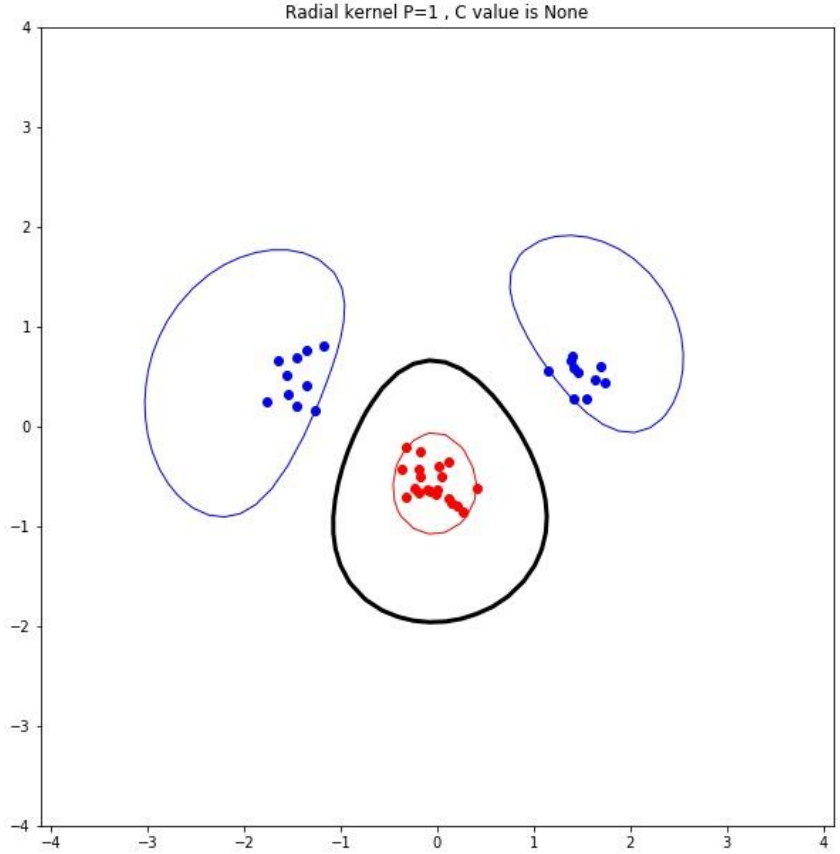
C controls the cost of misclassification on the training data. Small C makes the cost of misclassificaiton low, more slack Large C makes the cost of misclassification high, smaller slack

When you get an complex model which brings higher enough variance, then you want to make the model more general to fit more variaty data(lower noise in training data), then you can choose to increase slack(lower c) to get higher bias and lower variance.

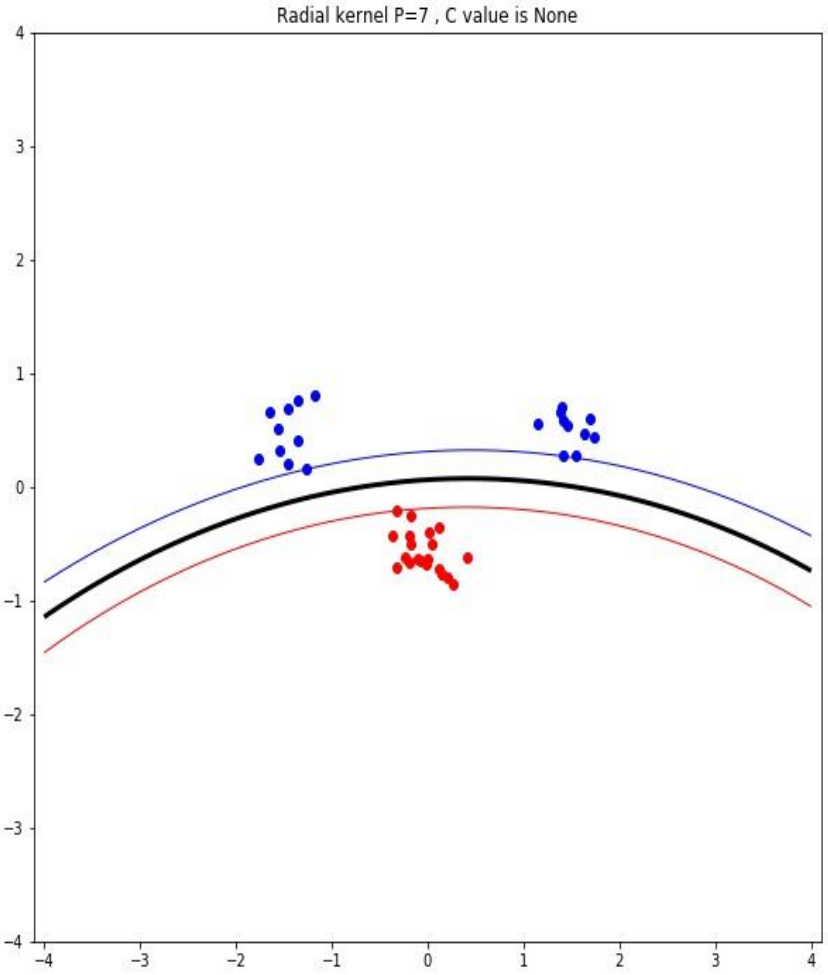
When you get an model with enough slack, the bias of it maybe too high, and this model maybe too simple, then you can increase the complexity of the model to improve the variance.

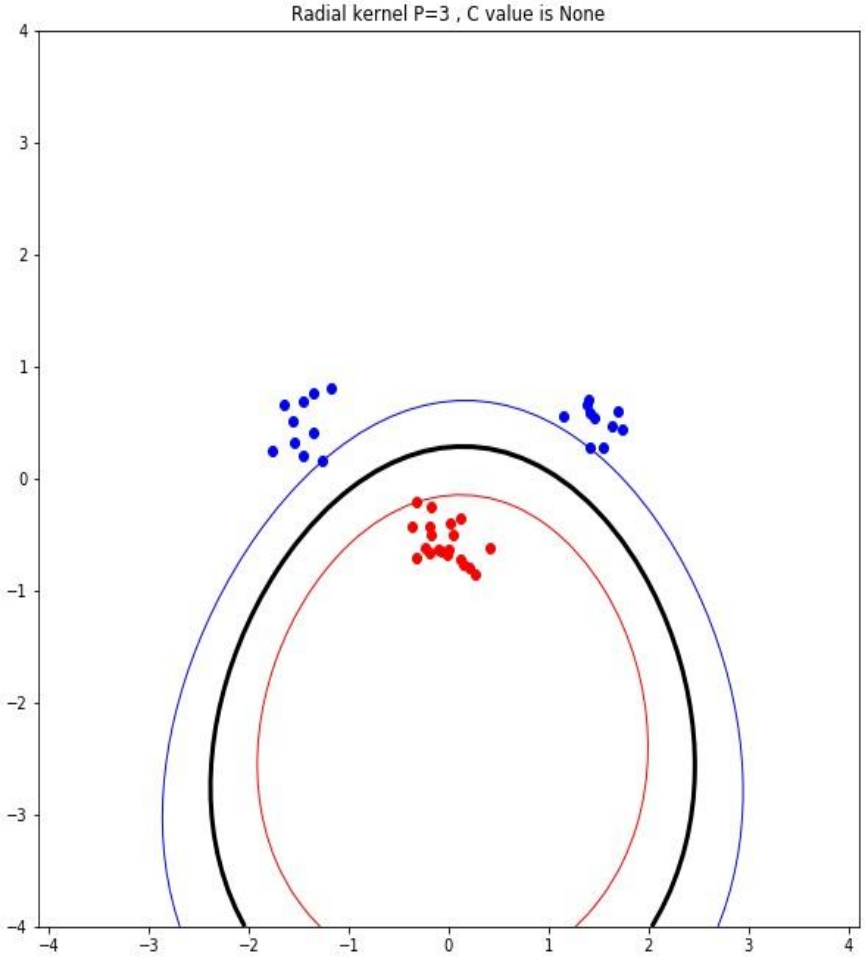
RBF Kernel using different theta = 2 and 1





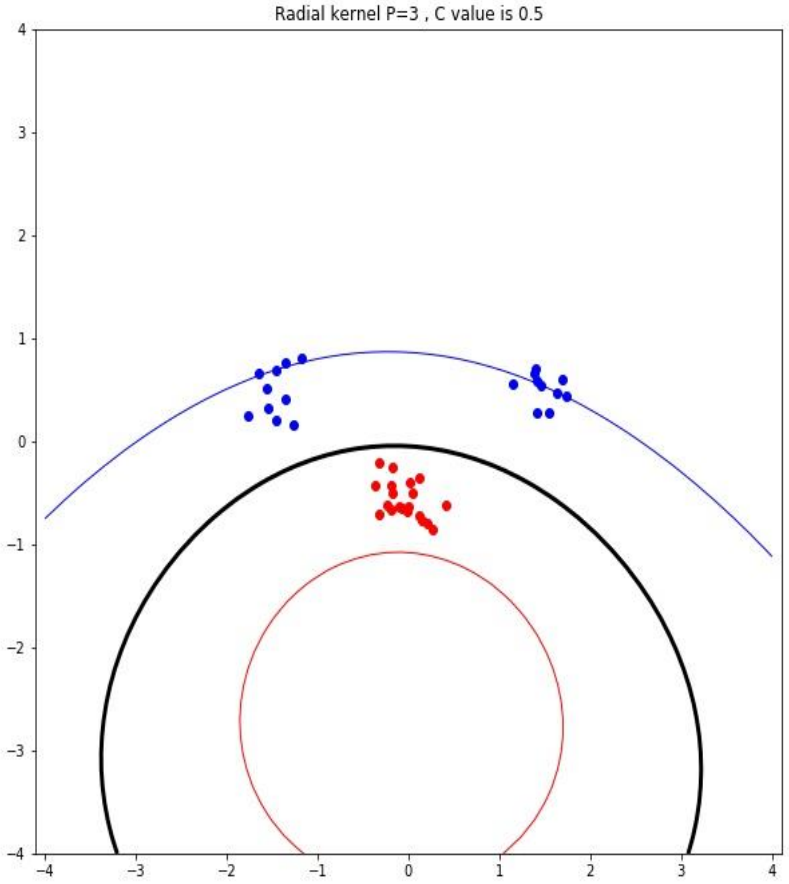
RBF Kernel using different theta = 7 and 3

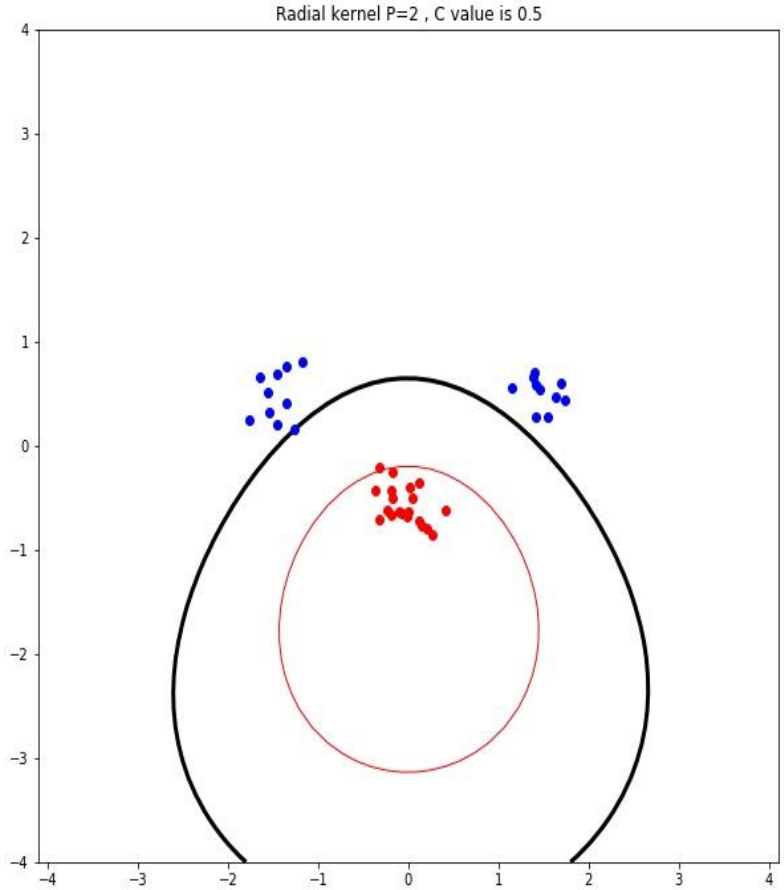




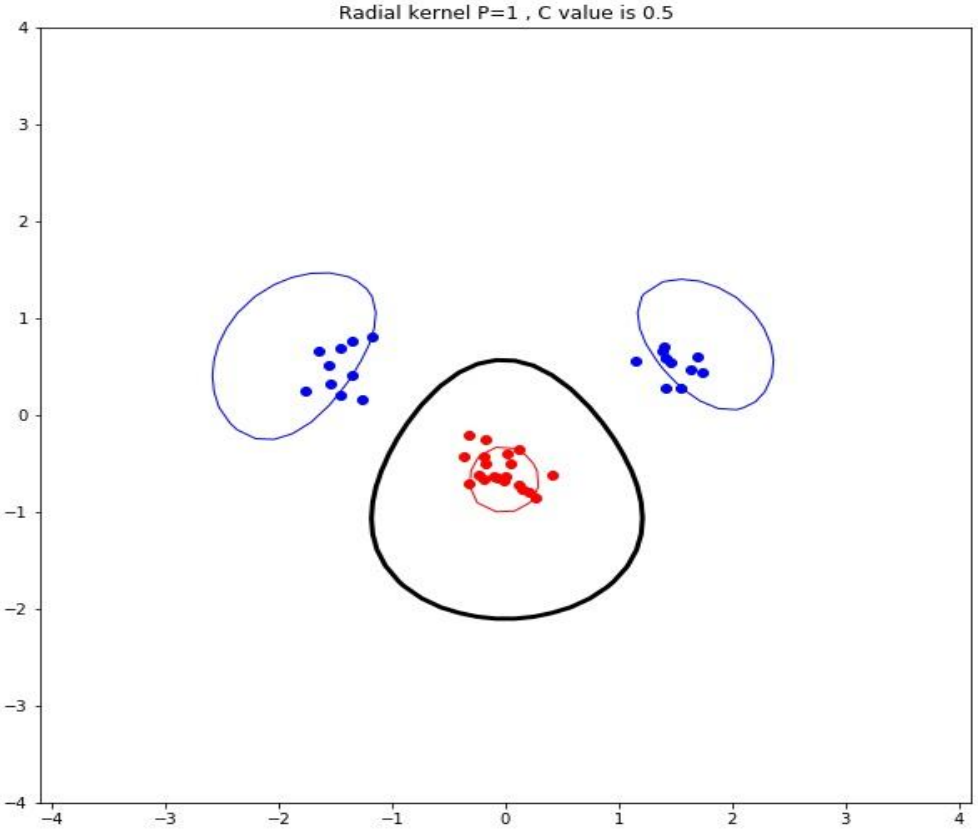
Introducing Slack:

RBF Kernel using different theta =



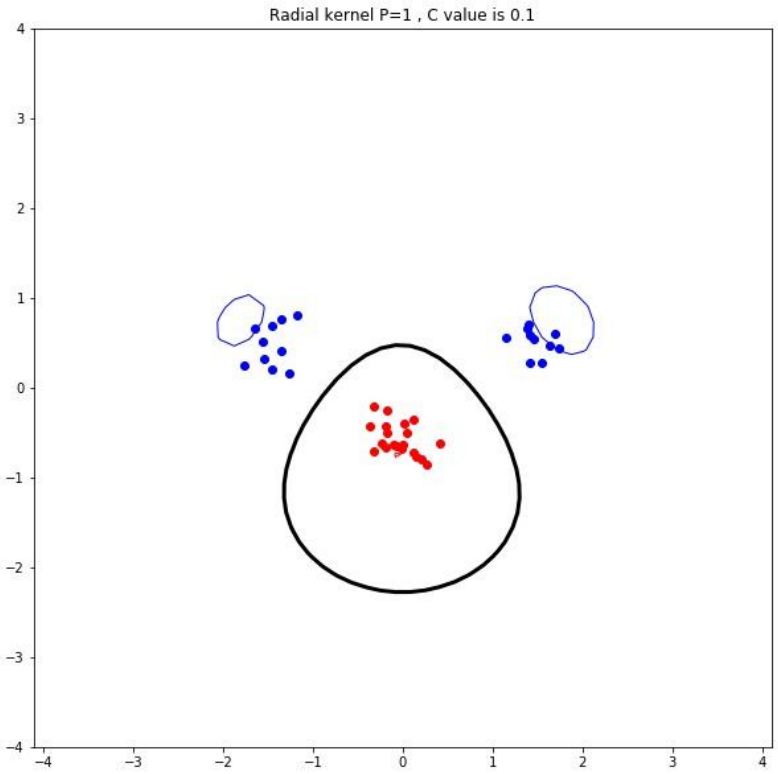


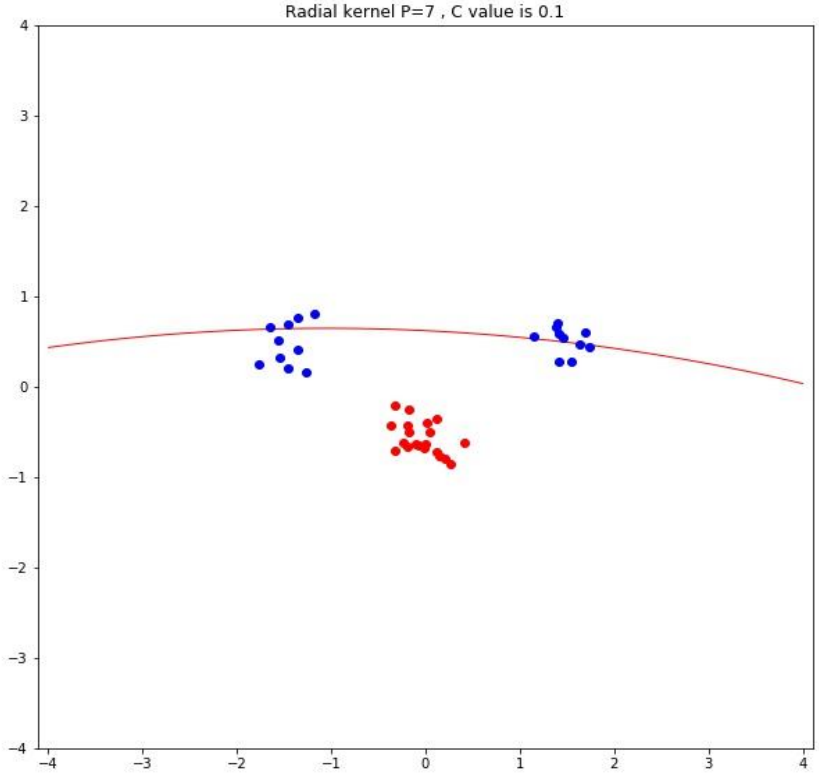
RBF Kernel using different theta = 0.5

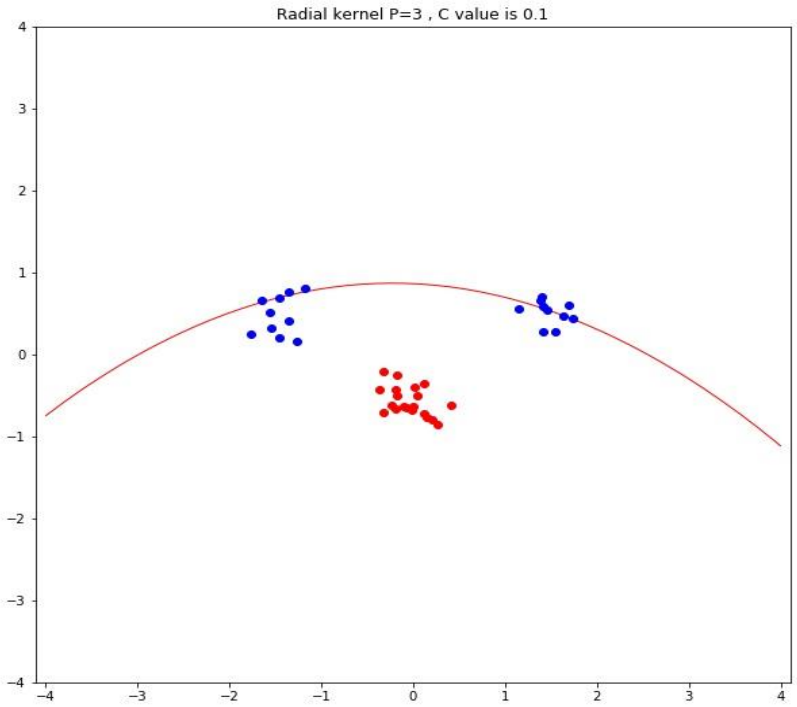


High bias low variance

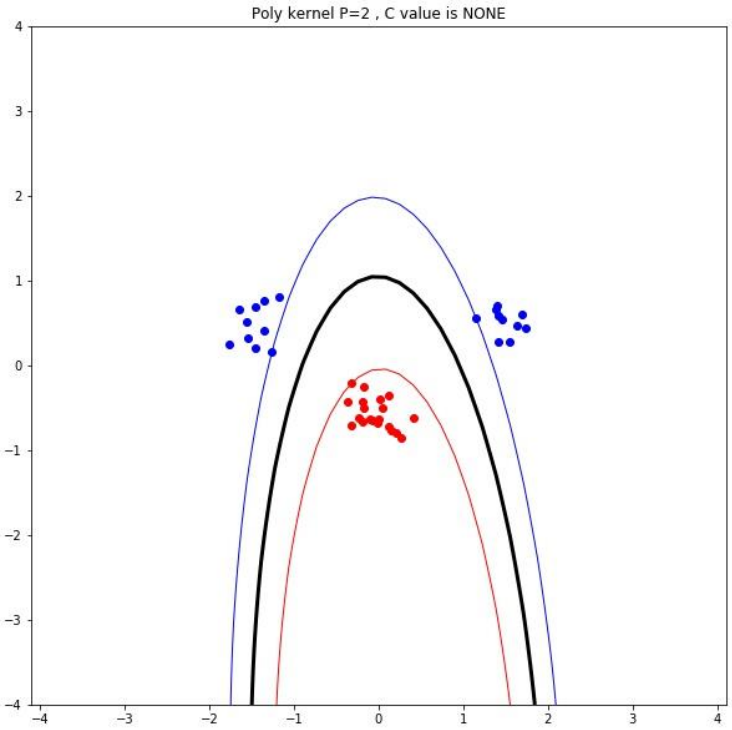
RBF Kernel using different theta = 1 , 7 and

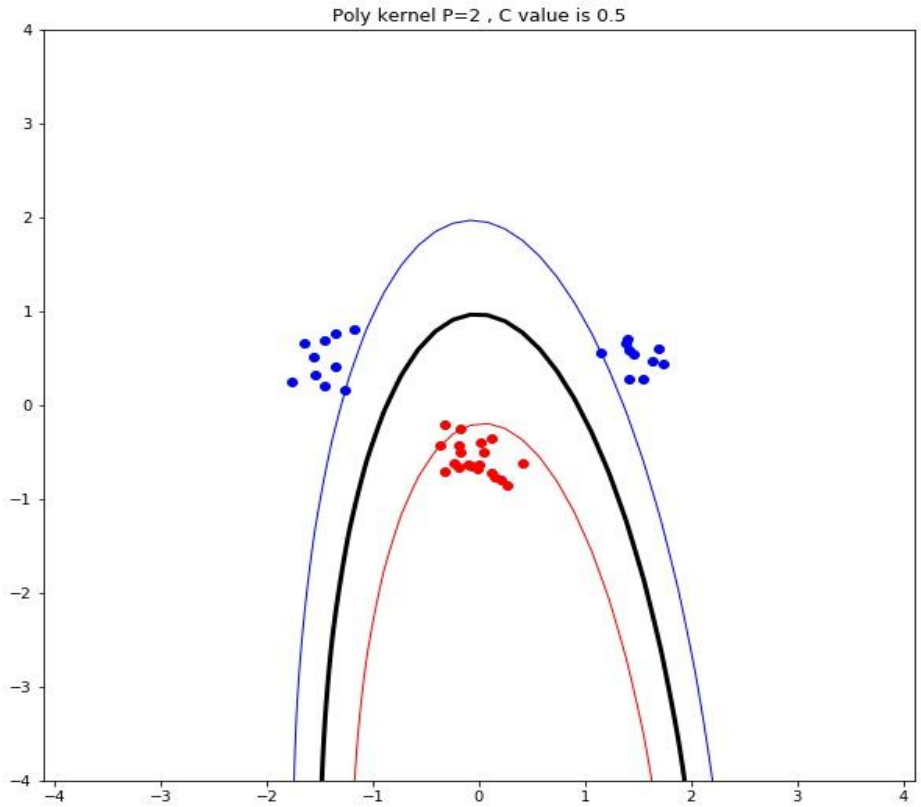


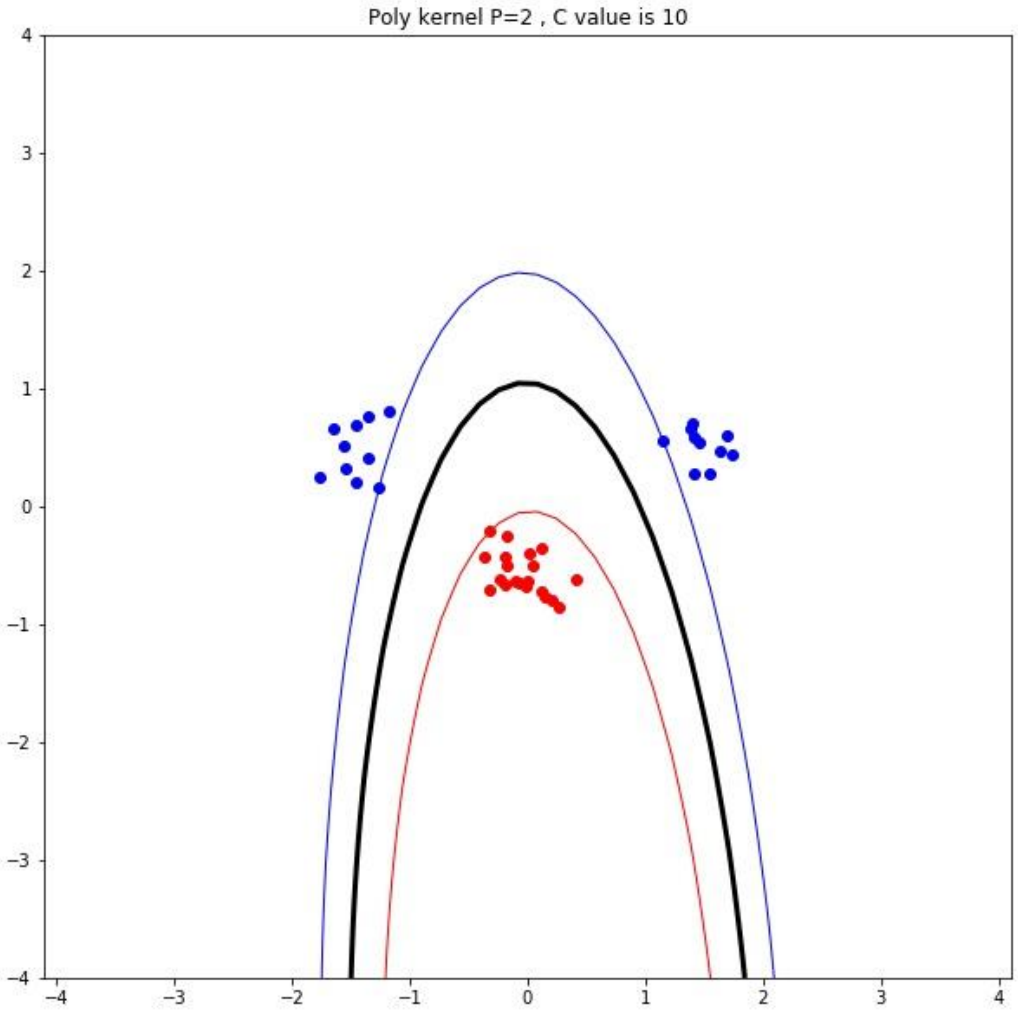


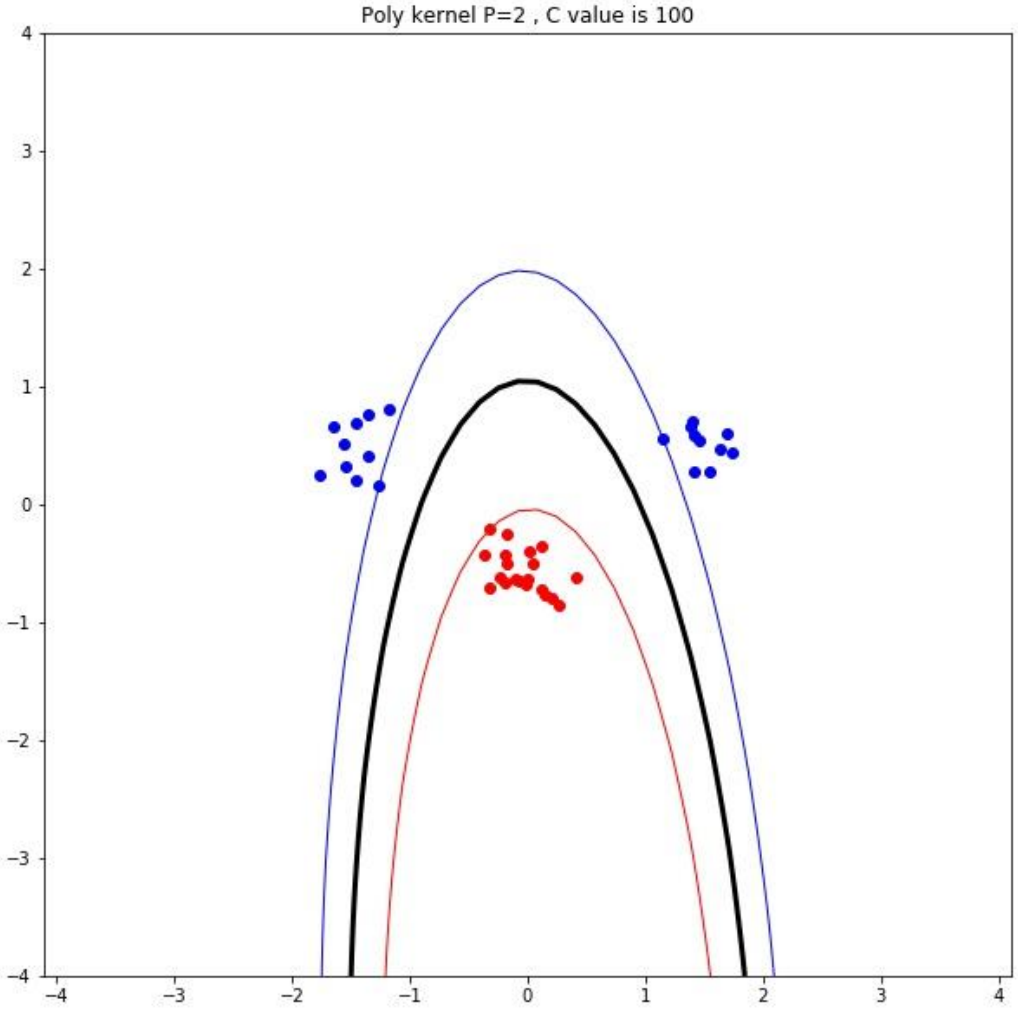

Radial kernel P=3 , C value is 0.1
4
3
2
1
0
−1
−2
−3
−4
−4
−3
−2
−1
0
1
2
3
4

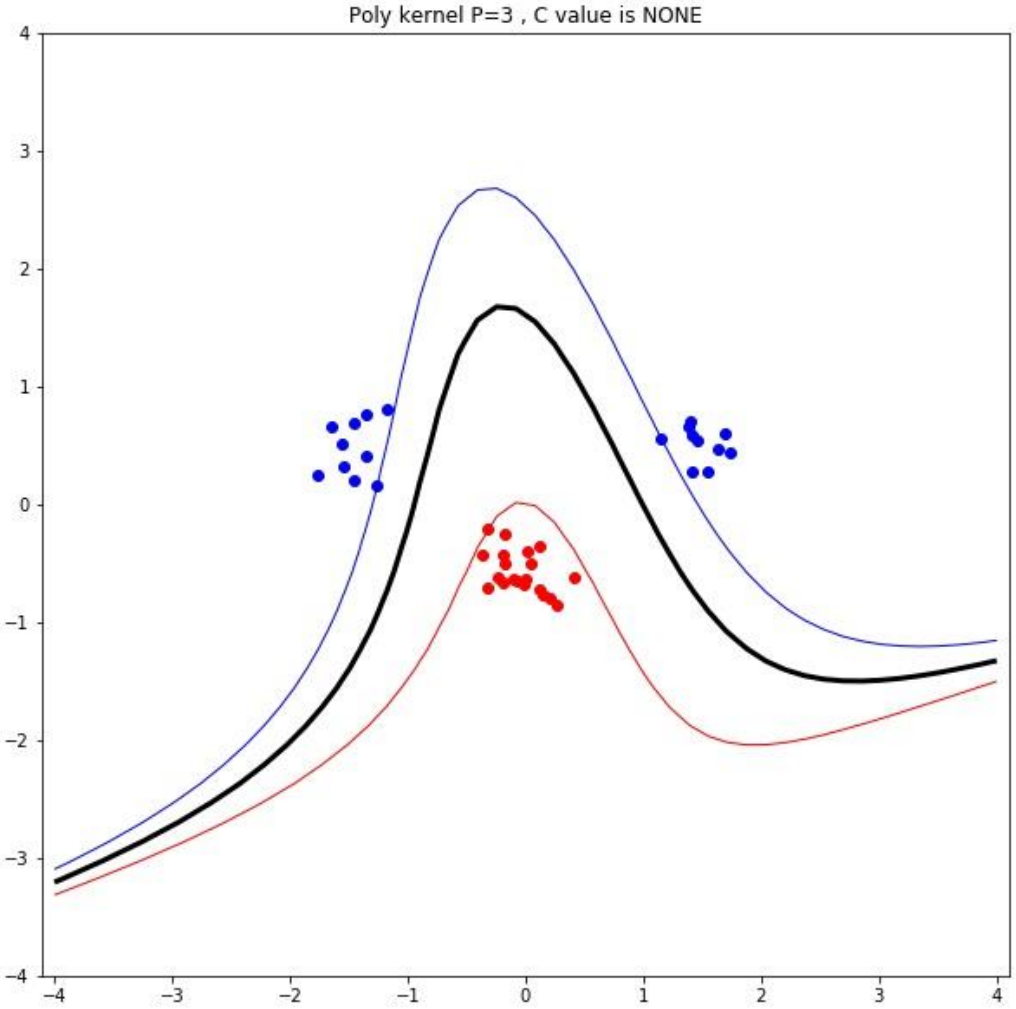
Poly Linear





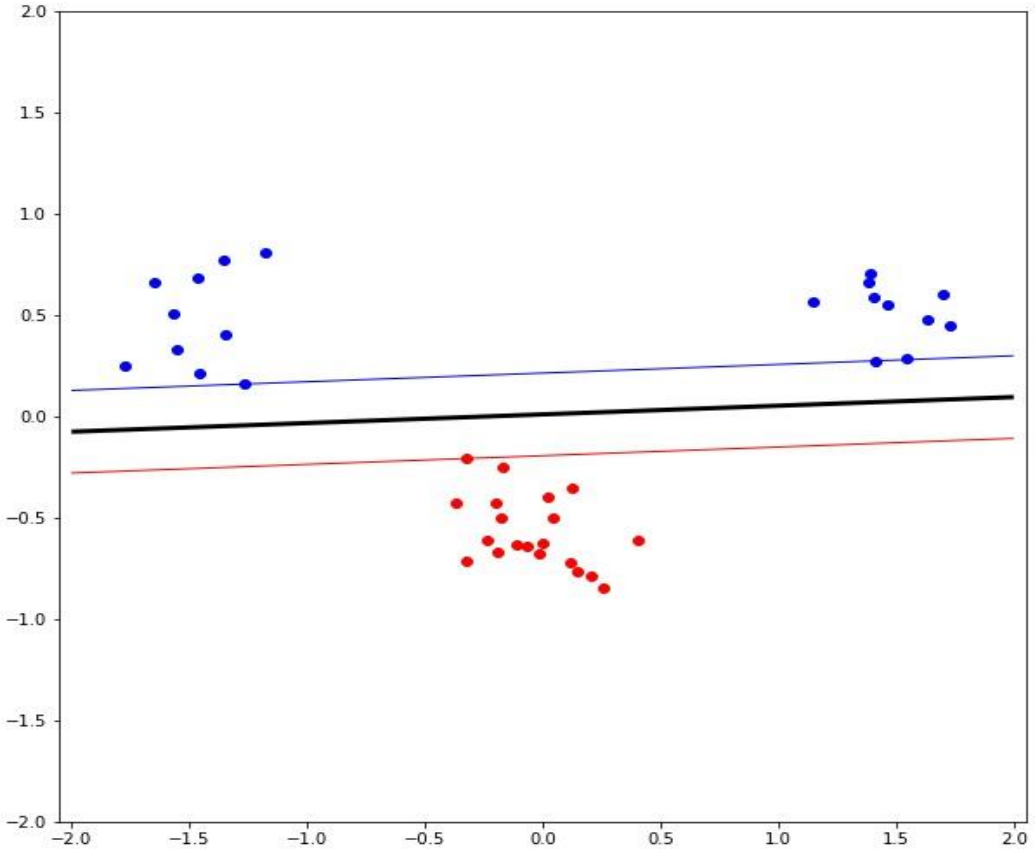

Poly kernel P=2 , C value is 10
4
3
2
1
0
−1
−2
−3
−4
−4
−3
−2
−1
0
1
2
3
4

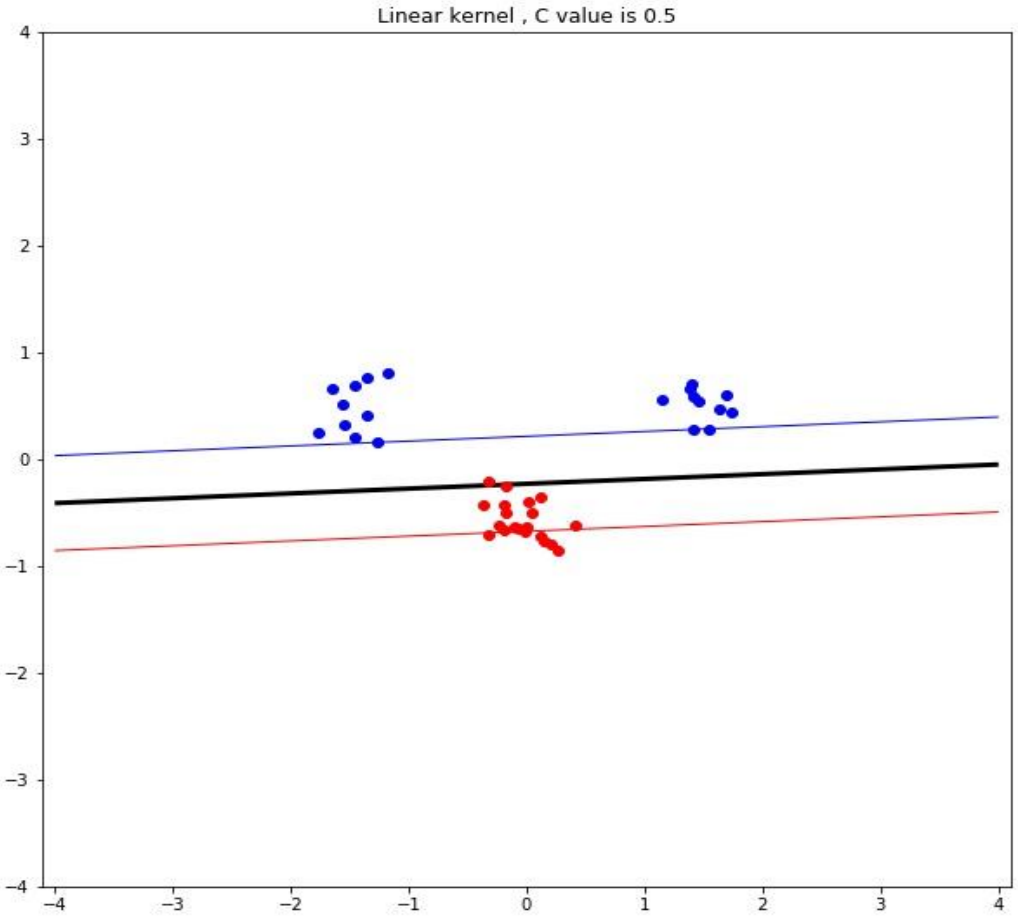

Poly kernel P=2 , C value is 100
4
3
2
1
0
−1
−2
−3
−4
−4
−3
−2
−1
0
1
2
3
4

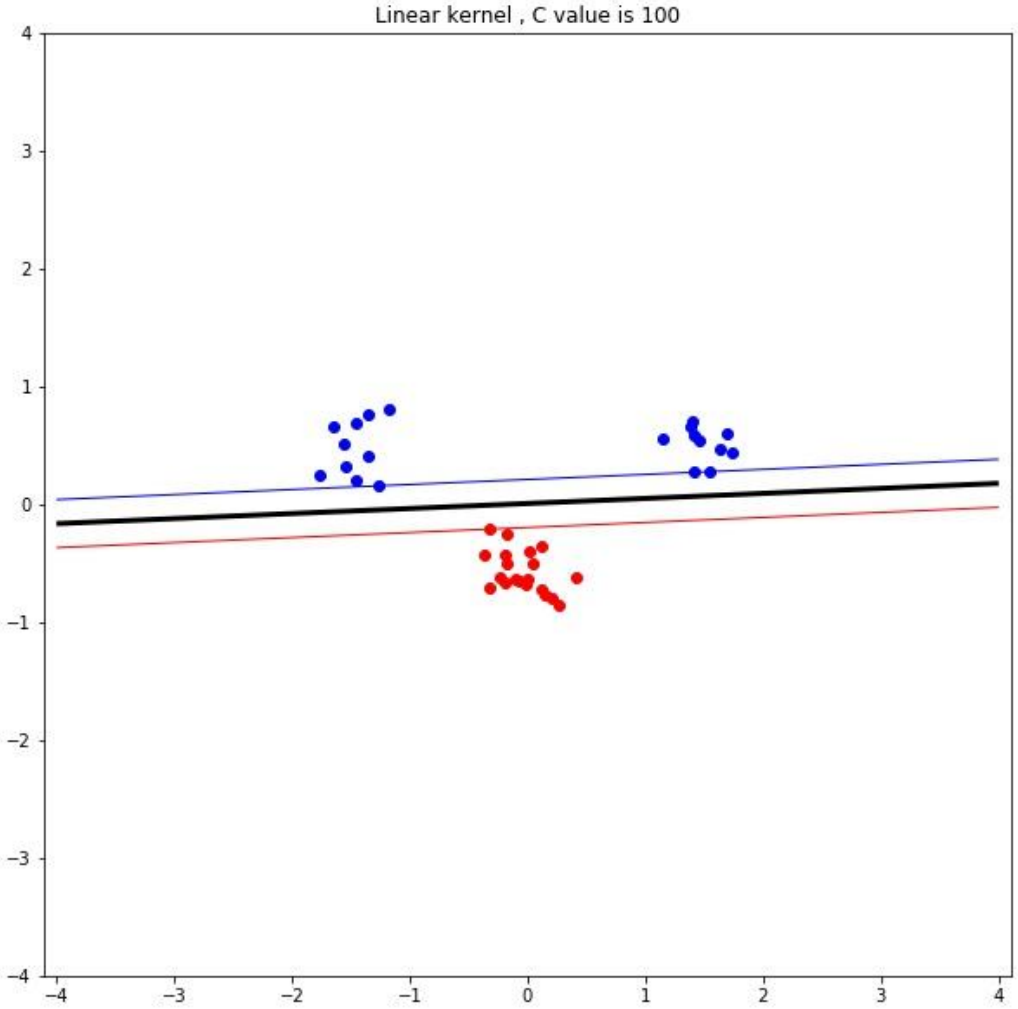

Poly kernel P=3 , C value is NONE
4
3
2
1
0
−1
−2
−3
−4
−4
−3
−2
−1
0
1
2
3
4

Linear K:

C = None




Linear kernel , C value is 0.5
4
3
2
1
0
−1
−2
−3
−4
−4
−3
−2
−1
0
1
2
3
4


Linear kernel , C value is 100
4
3
2
1
0
−1
−2
−3
−4
−4
−3
−2
−1
0
1
2
3
4