Nirmal kumar Ravi

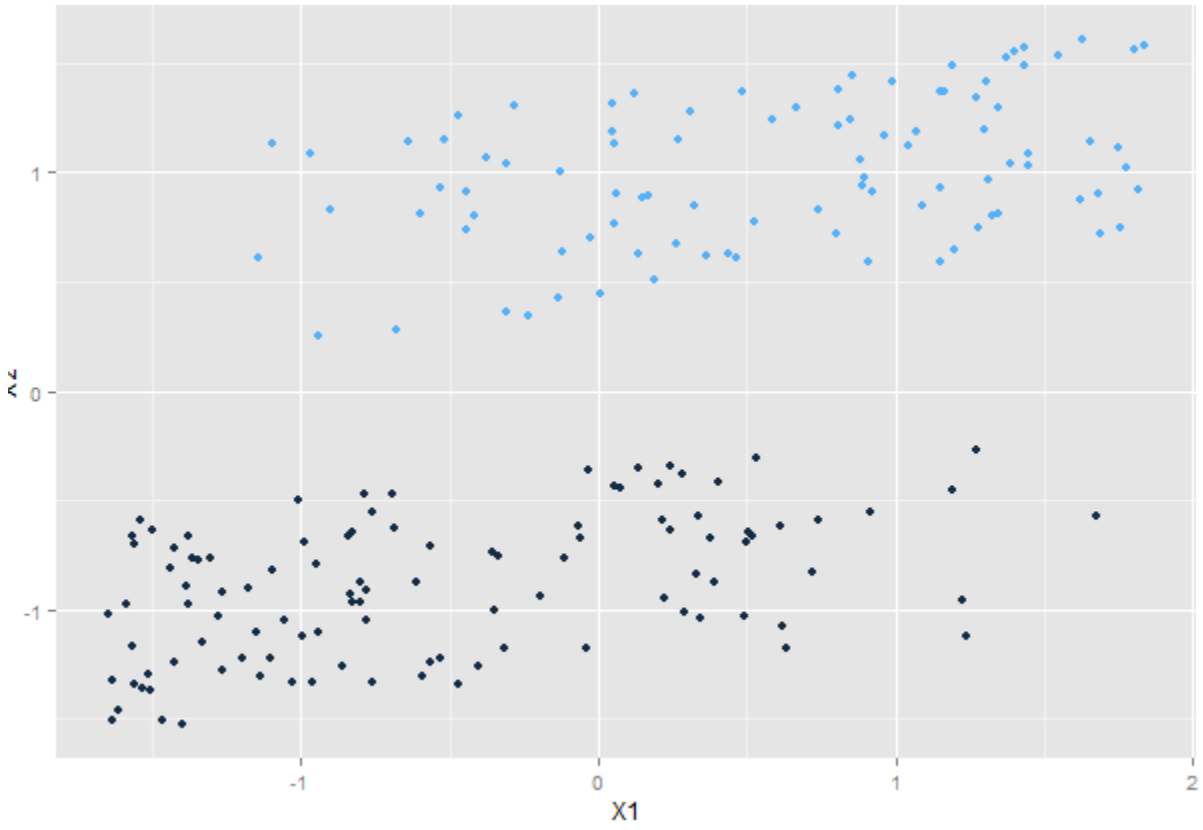
cs584 assignment 4

1. Generate a small dataset of 2D feature vectors of two classes such that the classes are linearly separable. Similarly, generate an additional set with examples that are not separable.

Data generation

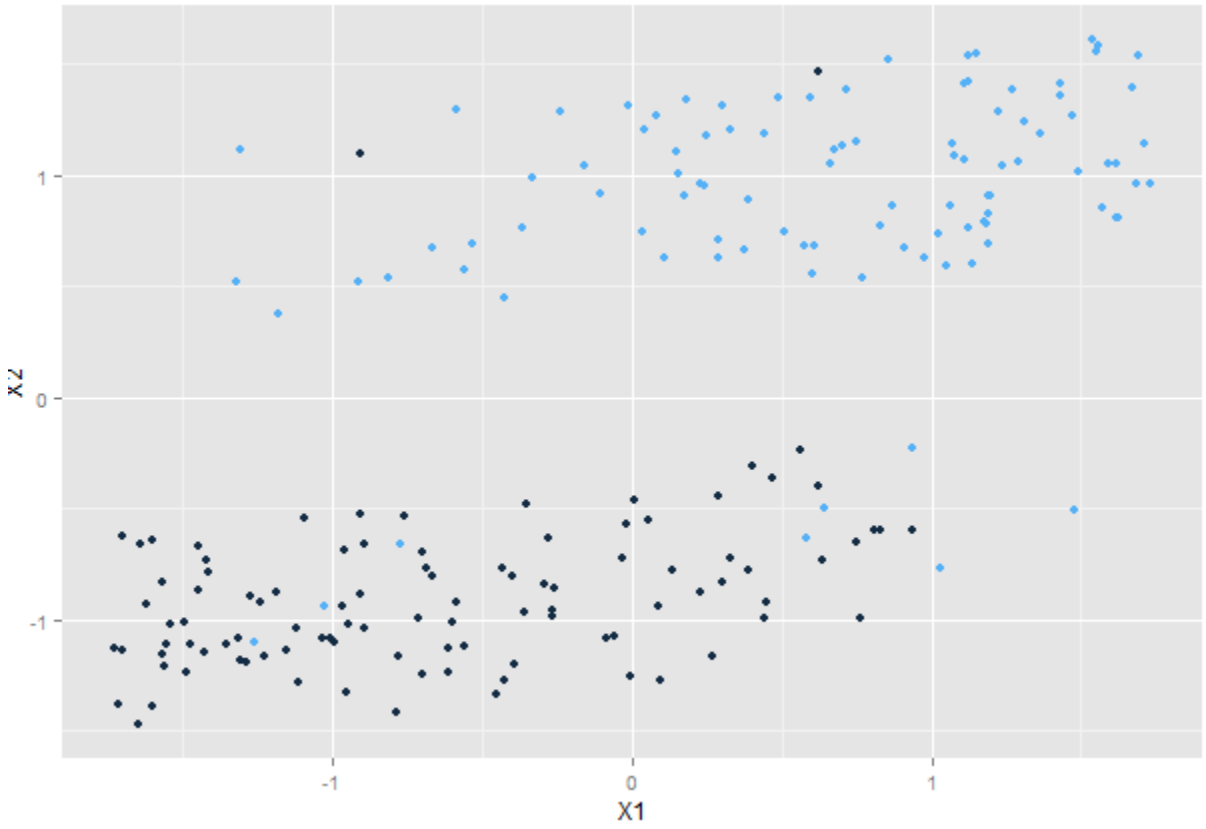
1)

Linear separable dataset



2)

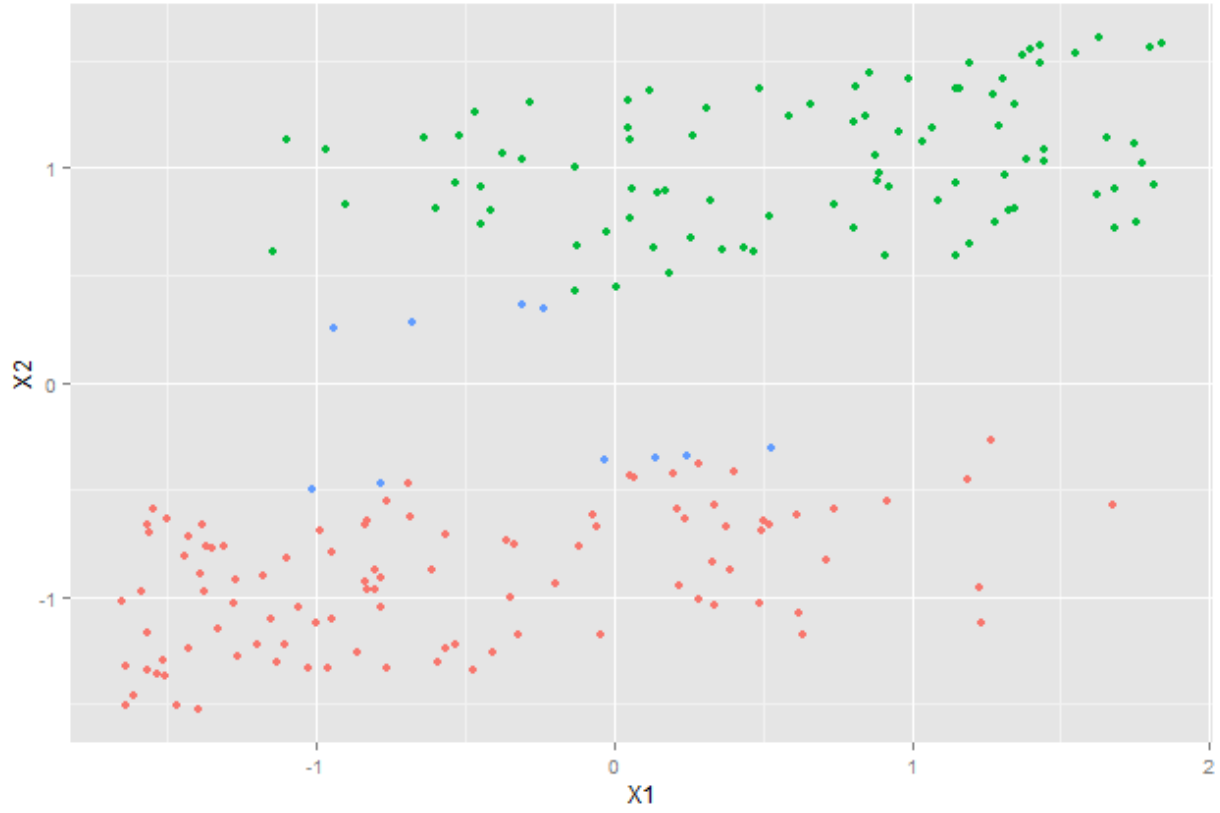
Not separable data set



2. Implement a linear SVM algorithm with hard margins and apply it to the separable dataset you generated. Plot the data points and mark the support vectors you identified. Apply the algorithm you implemented to the non-separable dataset and observe the results. Make sure to normalize the data if necessary (in this and subsequent algorithms you implement).

Implement hard-margin for separable data set

Support vectors plot



For linearly separable data, the hard margin works very well. The accuracy is 100%.

-0.3

W	820615 2.7188749
W0	-0.02281187

Implement hard-margin for non- separable data set

The algorithm suffers very badly. In fact it tries to fit data which is on other side of decision boundary.

3. The primal objective function of soft margins SVM is given by:

$$L_P = \frac{1}{2}\|w\|^2 + c\sum_{i=1}^{m}\xi_i - \sum_{i=1}^{m}\alpha_i(y^{(i)}(w^T x_i + w_0) - 1 + \xi_i) - \sum_{i=1}^{m}\beta_i\xi_i$$

Using the minimization equations $\frac{\partial L_P}{\partial w_i} = 0$, $\frac{\partial L_P}{\partial w_0} = 0$, and $\frac{\partial L_P}{\partial \xi_i} = 0$ develop the expression of the dual L_D that has to be maximized. You need to show and explain the development in addition to showing the final result.

Refer soln.pdf

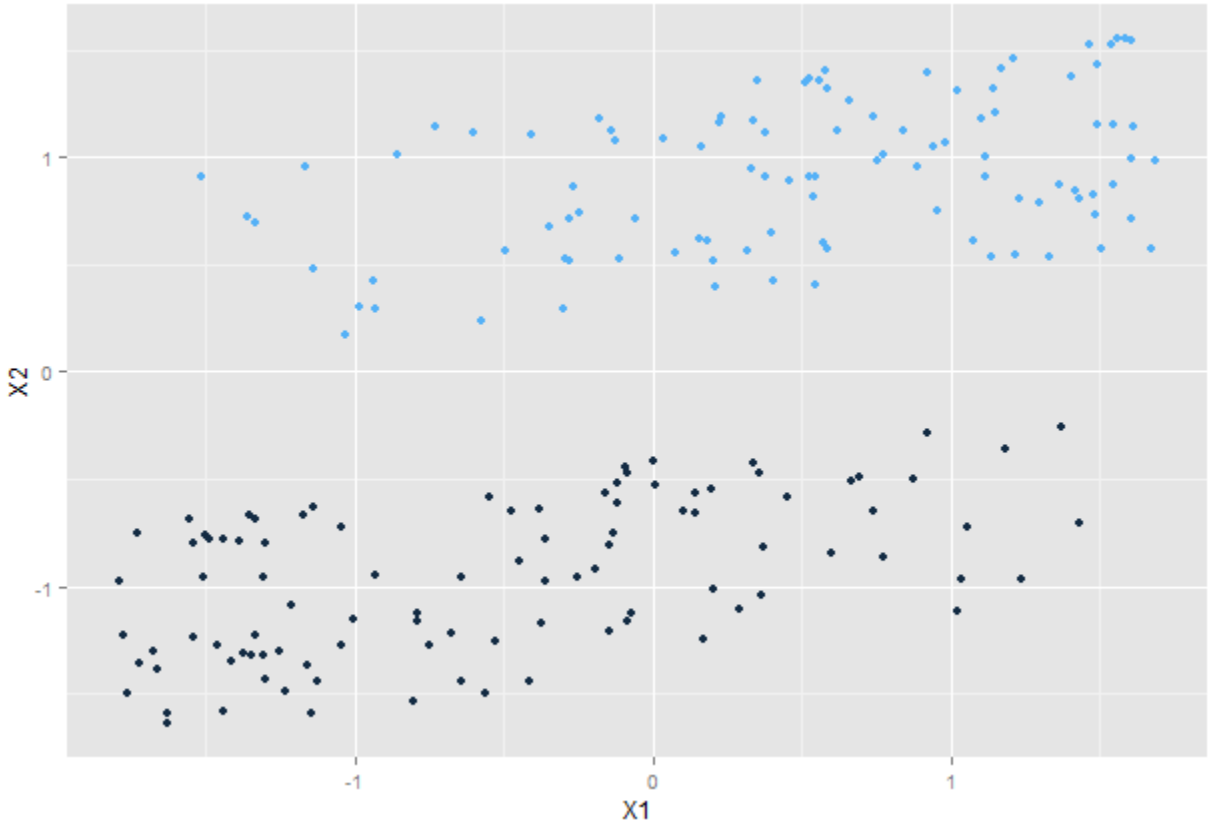
4. Implement a linear SVM algorithm with soft margins and apply it to the datasets you generated. Plot the data points and mark the support vectors you identified.

By soft-margin we were able to get 86% accuracy on non-separable dataset. Tried with different slack from 2^-10 to 2^20 and it seems 2^5 seems to be the best fit

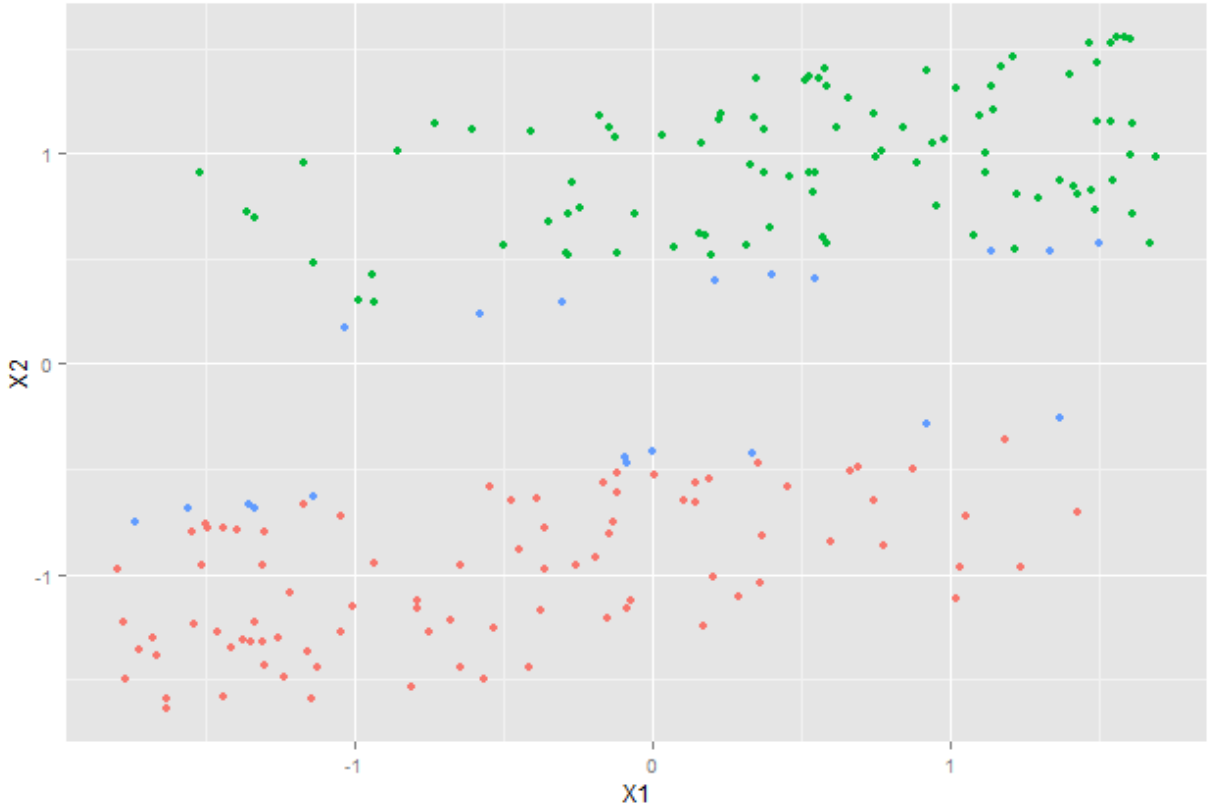
5. Implement a kernel-based SVM algorithm using a polynomial and Gaussian kernel functions. Apply your implementation to the datasets you generated. Plot the support vectors you obtain. Test the algorithm on two additional external datasets and report performance. Test and explain the effect of modifying different parameters of the algorithm.

Polynomial kernel

Data



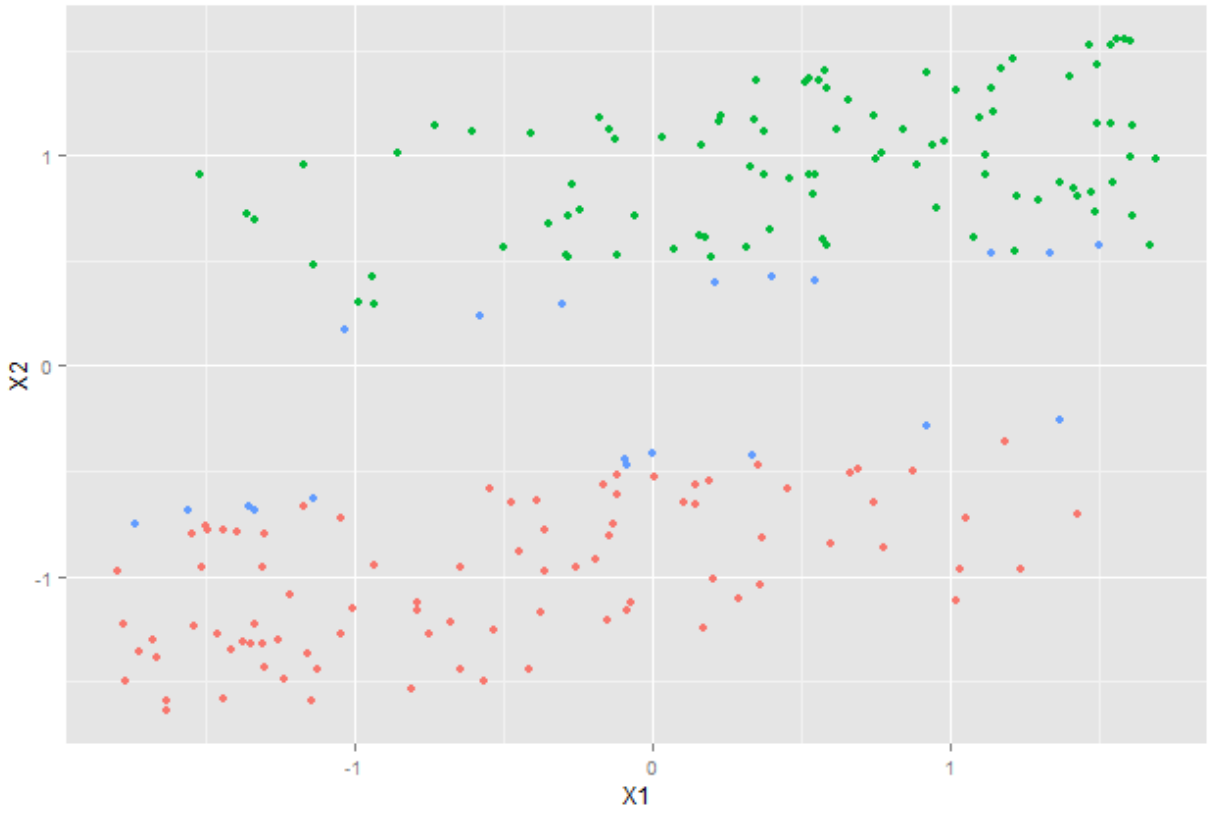
Support vectors



I tried different polynomial values and polynomial 2 seems to be the best fit for our data and accuracy is 100%

Gaussian kernel

Same with Gaussian kernel and with sigma as 5



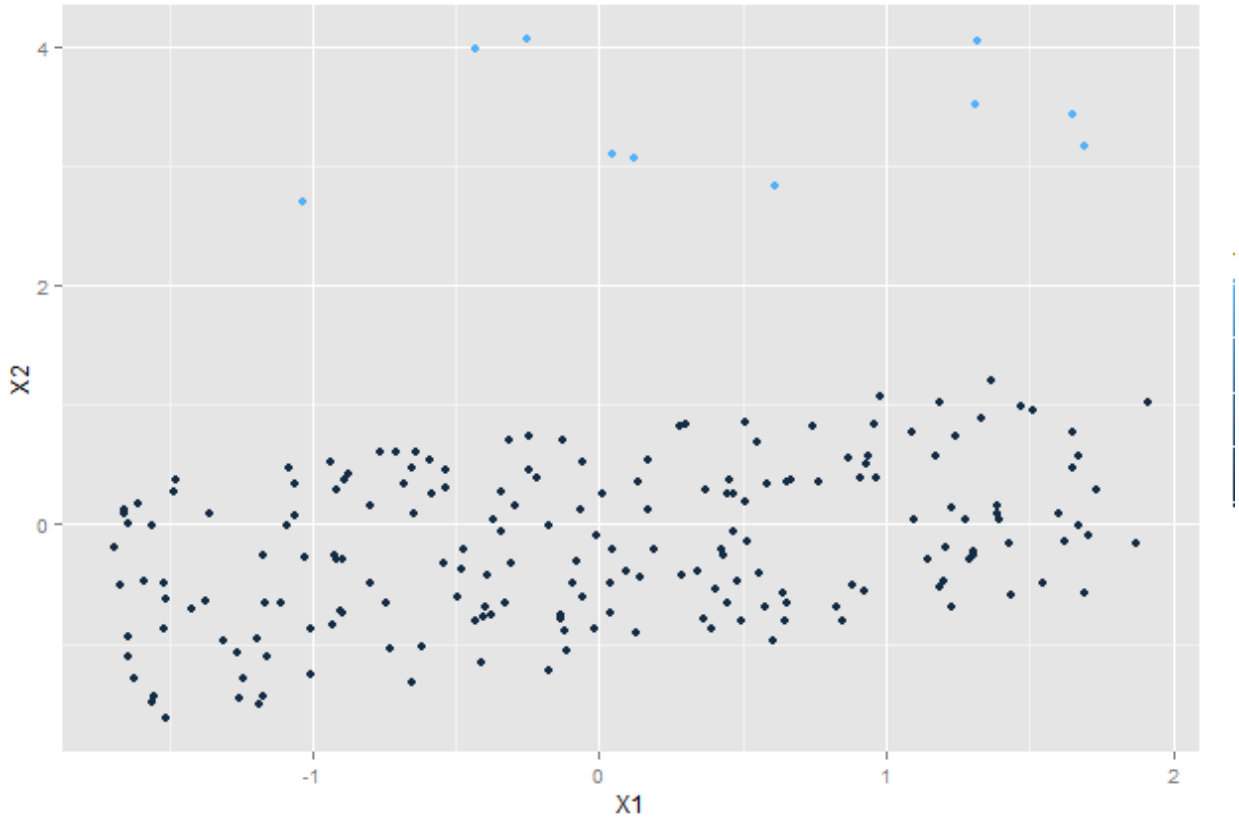
Let's try it on Iris dataset

Model	Additional parameters	Accuracy (%)
Linear		92
Polynomial	With polynomial 2	98
Gaussian	Sigma 4	83

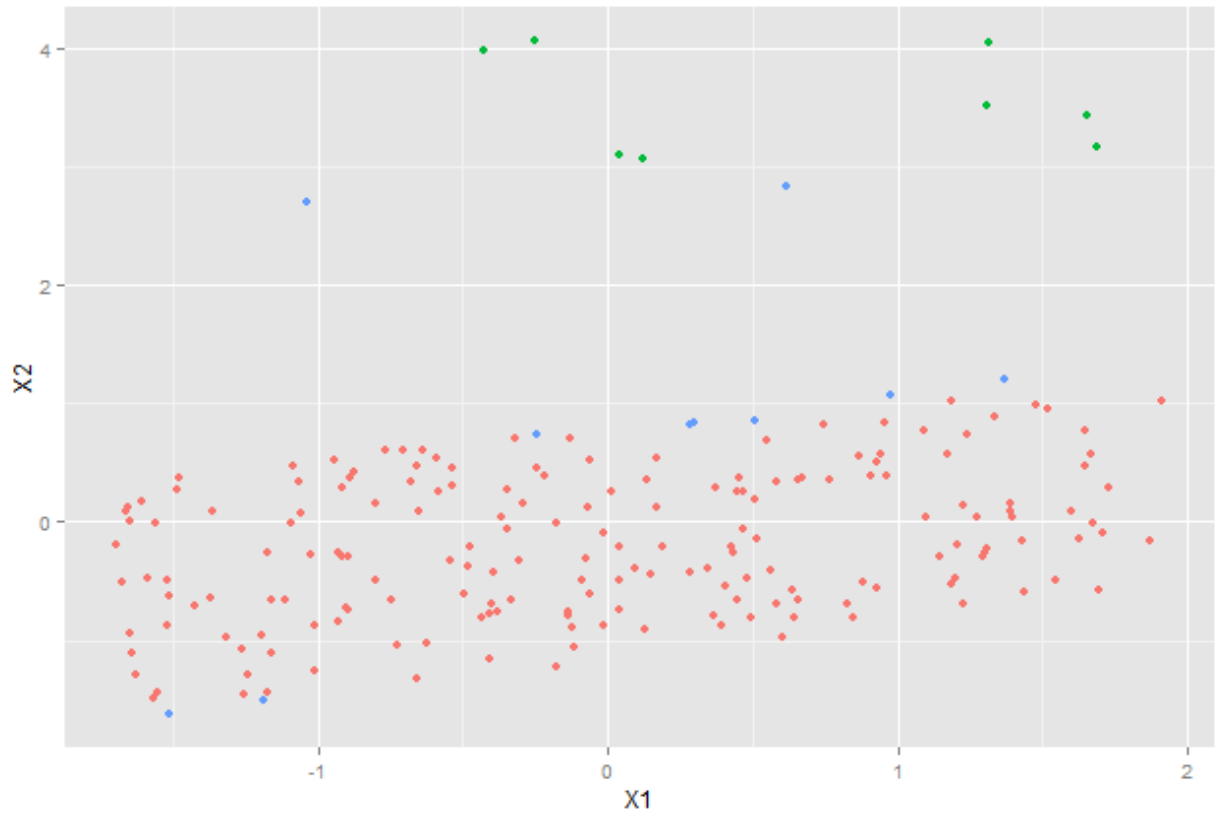
After fitting data on all models and doing 10 fold cross validation, The polynomial with order 2 seems to be the best fit

6. Test what happens when one of the classes has substantially more examples. Implement and test a solution for this case.

Data



In case of Dataset with more number of instances in one class. There tend to be more support vectors on class with more examples and less support vectors for class with lesser examples. In that case the decision boundary tend to lie closer to the data with more support vectors



As you can see from the figure there are more support vectors on class with greater number of instances. This can be mitigated by repetition of examples of the class with less data.