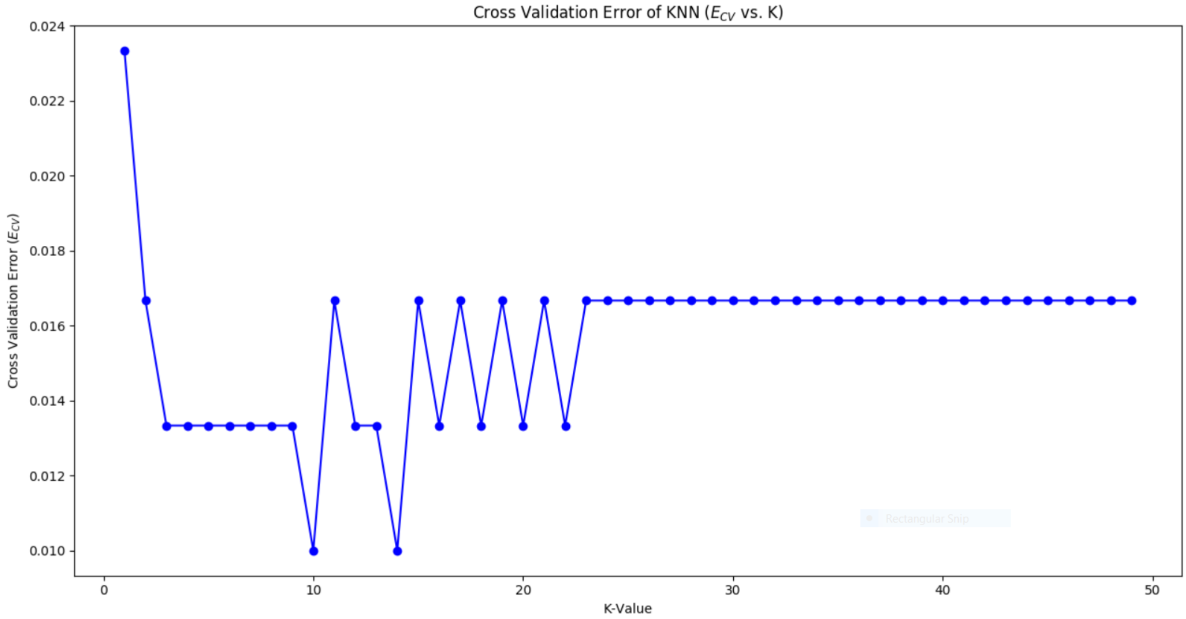
Assignment 11

Alberto Mejia
RIN: 661514960
CSCI 4100 - Machine Learning from Data
November 24, 2019

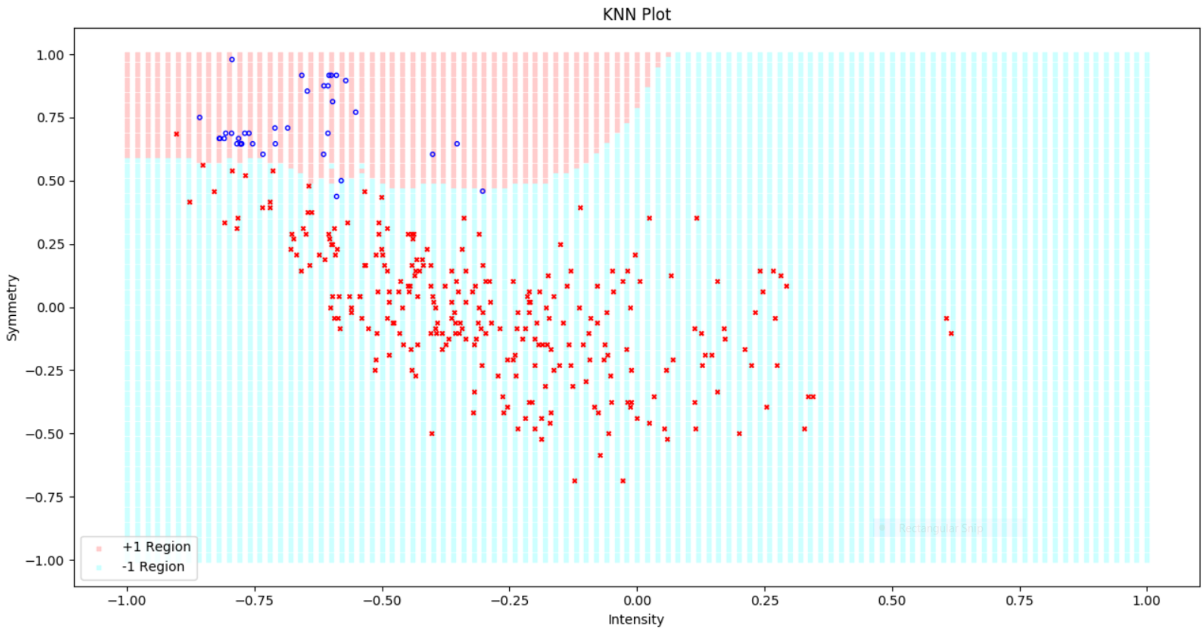
1. (450) k-NN Rule

(a) k-NN Rule



Optimal k: 10

(b) Decision Boundary

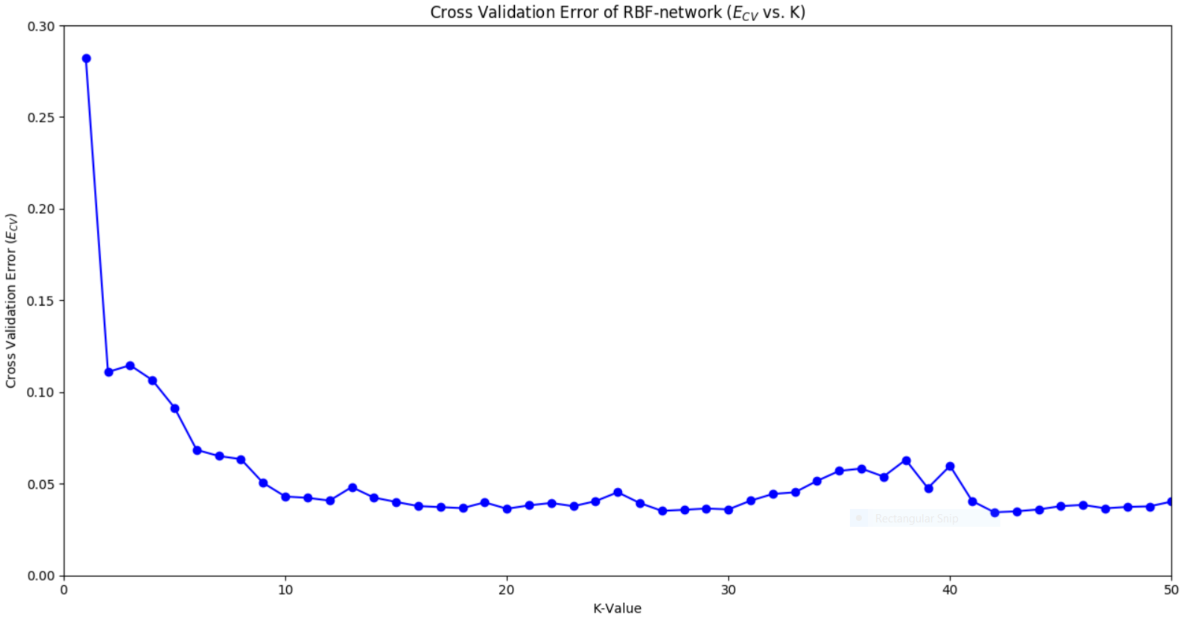


(c) Test Error

$E_{test} = 0.01467906$

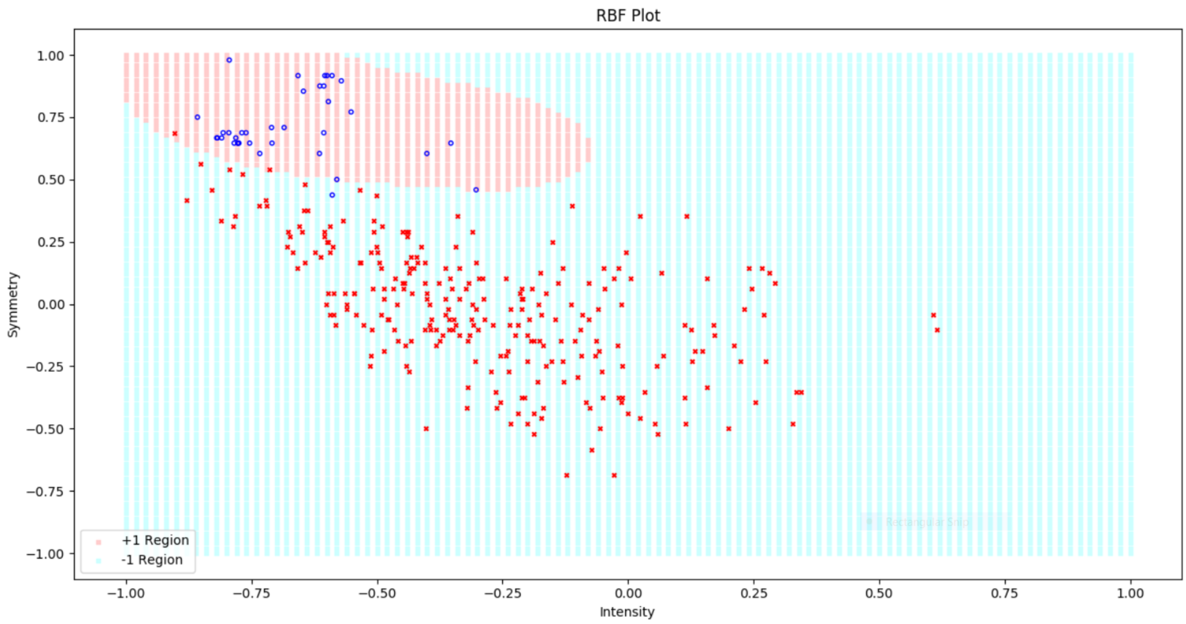
2. (450) RBF-network

(a) RBF-network with Gaussian kernel



Optimal k: 42

(b) Decision Boundary



(c) Test Error

$E_{test} = 0.010159$

3. (100) Compare Linear, k-NN, RBF-network

E_{test} for all three models is as follows:

(i) Linear Model: 0.01943

(ii) *k*-NN: 0.01467906

(iii) RBF: 0.010159

We can see that from these results, the test error of all three models are relatively the same. Thus, they all seem to be a good for the test data. RBF produced the lowest Cross Validation Error, and test error, of the three. It is not too surprising that RBF performed better than the linear model since it is essentially a linear model but with a linear transformation adjusted to the problem. This gives it more flexibility than a standard linear model. A representation similar to linear regression. RBF providing better results than *k*-NN is due to the large K value that RBF used. For RBF, we had an optimal k value of 42 whereas *k*-NN had an optimal k value of 10. This gave the RBF model a much tighter decision boundary, as we can see from above. The catch is that, k-NN has the lowest computation cost of the three. RBF has a higher space complexity; therefore, it uses a lot of memory and takes longer to run but has a very high accuracy. Choosing a model for this problem really depends on the users' needs for accuracy, memory consumption, and time complexity.