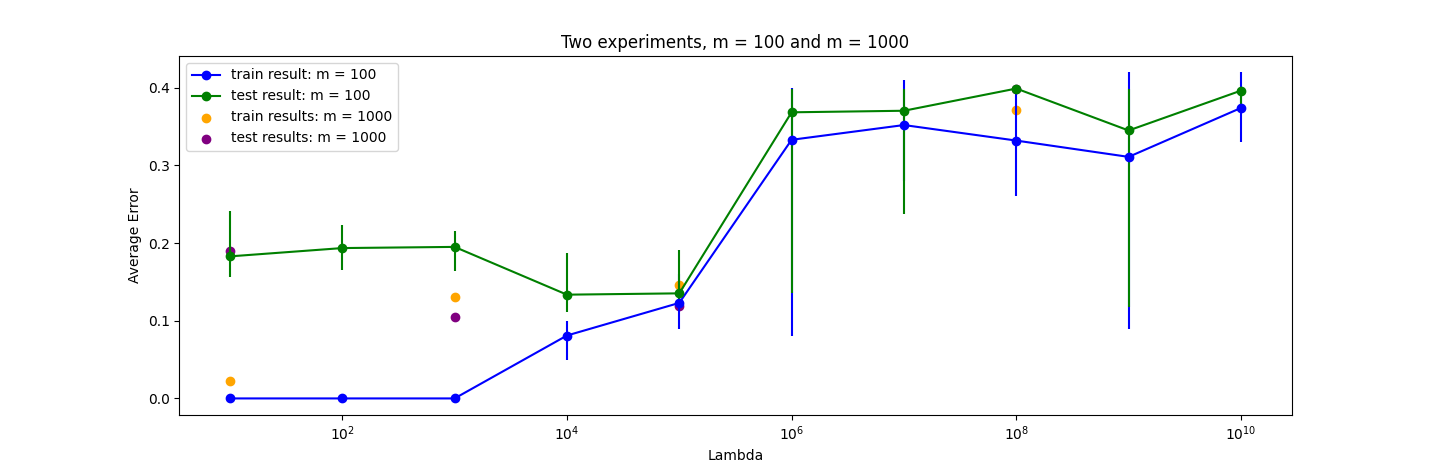
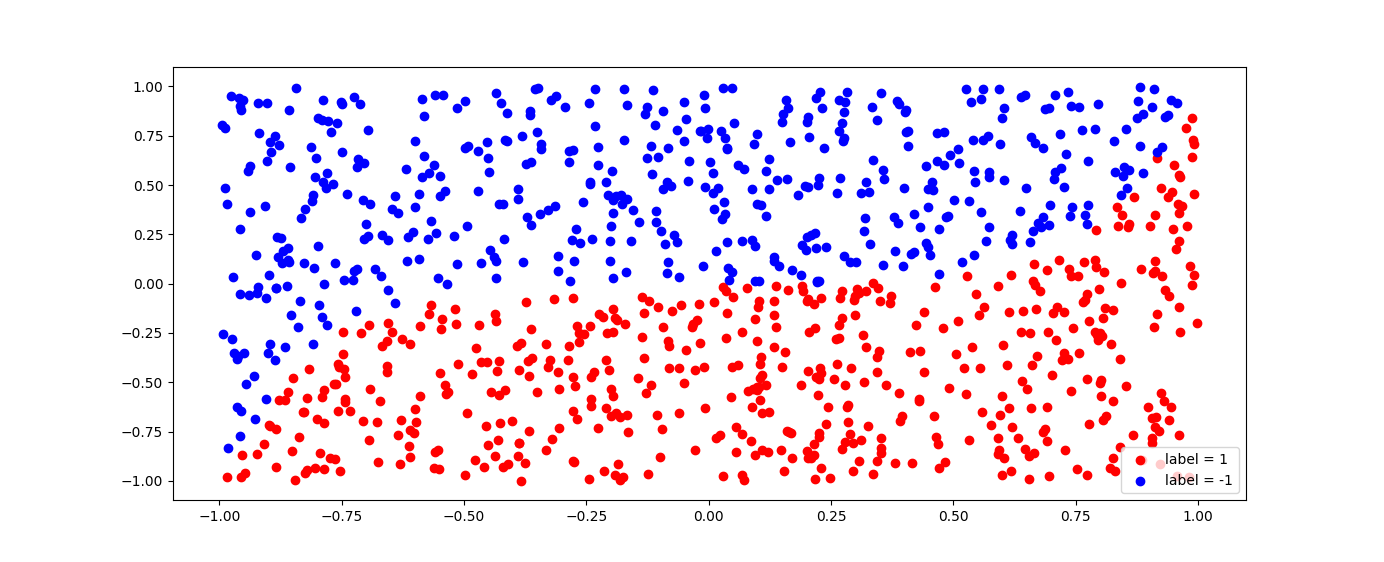
Q 2) a+b)



* Sample Size:
  + Training error: We expect to get a smaller training error with smaller sample size (in our case m = 100). The distribution might not be separable. Due to that, for a smaller sample size there is a higher chance to be separable. For a larger size of a sample, the sample is a better representative of the distribution, and has a higher chance of not be separable. This will cause us for higher error on the training sample.
  + Test error: We expect to get a smaller training error with larger sample size. As the sample size is higher, it is a better representative of the distribution and will result in lower estimation error.
  + Results: As shown in the plot above, the orange dots (higher sample size of the training set) are above the blue line (smaller sample size of the training set) as we expected. Moreover the purple dots (higher sample size of the test set) are below the green line (smaller sample size of the test set), or close to it as we expected.
* tradeoff:
  + Training error: We expect that as the increases, the training error will increase as well.   
    For small , we expect the SVM program to find a separator with small margin, which will result with smaller training error. As increases, we expect SVM to find separator with smaller norm, which will result in larger margin, which will result in higher approximation error.
  + Test error: We expect that as the increases, the test error will first decrease until the optimal value for , and then it will increase.   
    For small , we expect the SVM program to find a separator with small margin, which will result in high estimation error. When increases, we expect the test error to decrease until it meets the training error, where the optimal value is, where the tradeoff between the norm of the separator and the hinge loss is optimal. As continues to grow, we put more weight on the norm of the separator, and SVM will minimize the norm of the separator. This results in higher margin, which will cause us to penalize even correct labeled examples. This will cause a higher test error.
  + Our results: as we see in the plot above, the training error does increase as we expected, and the test error does decrease until it meets the training error, and then starts to increase.

Q4)

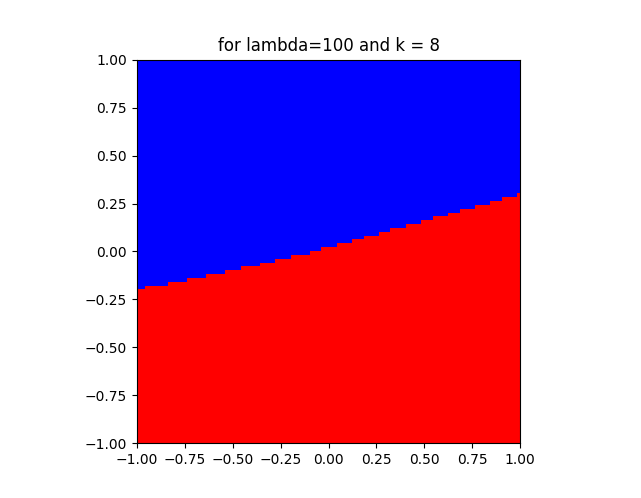
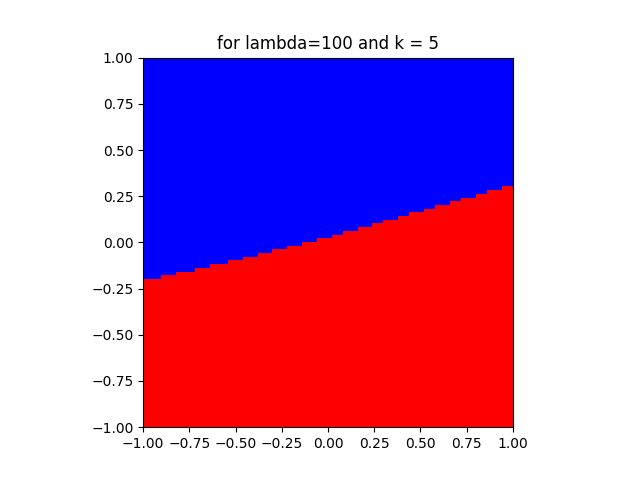
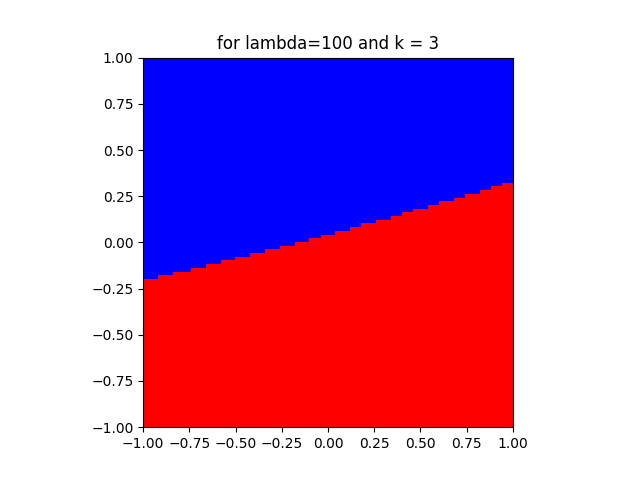
1.   
   according to the plot, it is easy to see that there isn’t a linear predictor that separates all the dots. But there is a polynomial separator that can separate all the blue and red dots better.

|  |  |  |  |
| --- | --- | --- | --- |
|  | 2 | 5 | 8 |
| 1 | 0.069 | 0.059 | 0.051 |
| 10 | 0.069 | 0.064 | 0.059 |
| 100 | 0.069 | 0.064 | 0.061 |

The best result was with with average error of 0.051.  
After running with the chosen parameters on the entire training set, we got error of 0.03.  
  
When running the 5-fold cross validation on the soft SVM without kernel we got:

|  |  |  |
| --- | --- | --- |
|  |  |  |
| 0.063 | 0.063 | 0.063 |

was chosen, and after running again with it on the entire training set we got error of 0.04.

1. The polynomial kernel SVM got a better error than the linear soft SVM. It is as we expected, as you can see from the distribution of the examples in in sub-question 4.a, there isn’t a linear separator, but there is a polynomial separator.
2. For a general classification problem, if the examples in the distribution are separable by a linear separator, then the linear soft SVM will result in better validation error than polynomial kernel soft SVM, as using the polynomial kernel might result in a new feature space that is not separable, which will cause it to return worse separator.  
   On the other hand, if the examples are not separable in the original space and are separable in the new feature space, then the polynomial kernel will return a better separator which will result in a better validation error.
3. 
4. The best for is .
   1. To calculate w we use: , where are our examples.

9)

1. By definition, .

We will prove that the functions in a, b are not kernel functions by showing that exists that for them

For we obtain:

By definition, .

For we get: .

In our case, for we obtain:

1. We will define as follows:  
   and for we get: