1. On a dataset with a clear non-linear boundary between its two classes, a linear SVM has both higher training error (20%) and test error (30%) than a radial SVM (0% error in both cases). It is clear that the linear boundary completely isolates only one of the two groups of the class that is divided by the other class, making it guaranteed to have a large number of errors. Conversely, the radial SVM perfectly captures the true boundary of the dataset, with absolutely no errors due to how well it matches the circular shape of the center class.
2. Non-linear logistic regression
   1. Generated dataset
   2. Plotted observations
   3. Logistic Regression
   4. Training plot
   5. Non-linear Logistic Regression
   6. Training plot
   7. SVC with plot
   8. SVM with plot
   9. The support vector machine and non-linear logistic regression both made little to no errors on their predictions for the training set, while the logistic regression and support vector classifier both performed quite poorly even on the training set. This is obviously because of the fact that the true boundary between the classes is quite clearly non-linear.
3. A barely linearly separable test
   1. Generated data
   2. The cross-validation errors were as follows:
      1. Cost=0.001 -> 32 errors
      2. Cost=0.01 -> 3 errors
      3. Cost=0.1 -> 2 errors
      4. Cost=1 -> 0 errors
      5. Cost=5 -> 1 error
      6. Cost=10 -> 1 error
      7. Cost=100 -> 1 error

The training errors match this somewhat, with the first two cost values retaining much higher errors than the rest. However, cost=100 had the least training error, but cost=0.1 had the least cross-validation error, likely due to overfitting.

* 1. The value of cost that leads to the least test error is 10. This is different from the optimal value chosen by both training error and cross-validation. The training error predicted the largest given value of cost=100 to have the least error, but the test error refuted this and chose a somewhat smaller value of 10 instead. The cross-validation predicted a lower optimal value of cost than the test error did.
  2. The accuracy of this experiment may have benefited from using a larger dataset, as the one used here had n=50, which may have been a little low.

1. Auto
   1. Created binary variable
   2. Cross validation errors were as follows:
      1. Cost=0.001 -> 0.135
      2. Cost=0.01 -> 0.089
      3. Cost=0.1 -> 0.097
      4. Cost=1 -> 0.095
      5. Cost=5 -> 0.099
      6. Cost= 10-> 0.112
      7. Cost=100 -> 0.125

The training errors for each of the respective costs seems to match up with the cross-validation errors overall, although cost=0.1 had the lowest training error while cost=0.01 had the lowest CV error.

* 1. Both the polynomial training errors and CV errors were very high, except for with the CV error with degree 1, which amounts to a linear boundary. Errors for higher degrees were substantially higher than that of a polynomial kernel with degree one. On the other hand, the radial training errors and CV errors were very low, with the best CV results coming from a gamma value of 1 and a cost value of 0.001, and the best training results coming from a gamma value of 10. Overall, the radial kernel support vector machine seems to have performed significantly better than the support vector classifier, which in turn did significantly better than the polynomial kernel support vector machine.
  2. Keep getting error when trying to plot, but only with auto dataset. Fix later at some point.

1. OJ
   1. Split
   2. The support vector classifier created has 439 support vectors, which is a fairly large number given that the OJ dataset contains only just over 1000 observations. The support vector distribution is almost perfectly half-half, with 219 in the CH class and 220 in the MM class.
   3. The training error rate for the SVC was about 16.9%. The test error rate was 15.2%.
   4. Optimal cost is 0.01.
   5. With the new optimal cost, the training error was 17%, and the test error rate was still 15.2%.
   6. Repeat above but with polynomial kernel. First model’s training error was 16.5%, and its test error rate was 18.1%. The optimal polynomial model has degree=1 and cost=10. It has a training error of 16.25%, and a 15.6% test error rate.
   7. Repeat above but with radial kernel. First model’s training error was 11.1%, and its test error rate was 19.6%. The optimal radial model has gamma=0.01 and cost=1. It has a training error of 16.5% , and a test error of 15.6%.
   8. Overall, the approach that seems to give the best results on the OJ dataset is the linear support vector classifier. While the radial support vector machine appears to have lower training error, even the optimal radial model has a test error rate that is slightly higher than the optimal linear model, which seems to indicate that the latter is superior. The polynomial model performs worse than the former two.