Three models - KNN, RF, and SVM - had tuning parameters.

KNN’s tuning parameter was k which represents the number of observations that the model is told to use to determine which class a new observation is assigned to. My code specified a tuneLength of 10, which resulted a model being fit for ten different k values. The following values were tried for k: 5, 7, 9, 11, 13, 15, 17, 19, 21, 23. The k with the highest accuracy, k=7, was selected as the best fit and used for the final mode.

RF’s tuning parameter was mtry. This parameter represents the number of variables chosen at each sample split as the trees are created. Due to there only being three predictors, values of mtry=2 and mtry = 3 were considered. The model fit with mtry=2 produced a higher accuracy than mtry=3 and was selected as the best fit. It should be noted that difference in accuracy between mtry=2 and mtry=3 incredibly small (hundred thousandths place). I would also like to comment that the parameter ntrees, the number of trees built to fit the model, can also be specified for the model. Caret automatically chose 500 trees. Based on the high accuracy produced by 500 trees I did not investigate this parameter further.

SVM has two tuning parameters C, the cost penalty (which ultimately controls the margin size), and sigma, a scaling factor for the exponential behavior (which referred to as gamma in our textbook). More specifically, this sigma parameter is called the “inverse width parameter in the Gaussian Radial Basis kernel”. The larger your C value the bigger the penalty for misclassifications. This results in fewer support vectors and a smaller margin. Small margins result in lower bias and higher variance. With caret a tuneLength = 10 was run. Ten values for C were fit on different models. Trying to explain how sigma was chosen, I realized only one sigma value was used when tuneLength is called. [This thread]( <https://stackoverflow.com/questions/38859705/r-understanding-caret-traintunelength-and-svm-methods-from-kernlab>) indicates that svmRadial chooses a single sigma based on the output from [sigest]( <https://www.rdocumentation.org/packages/kernlab/versions/0.9-29/topics/sigest>) a function from the kernlab library. Per the function’s description, sigest returns a range of values for sigma which will return good results. I assume svmRadial chooses one of those sigma values, on what basis this decision is made I’m not sure. If I had realized this sooner, I could have used svmRadialSigma which tunes over both cost and sigma.