MSiA 400 Lab Assignment 2: mjk3551

Matt Ko

Problem 1a

Split the data into testing, training, and validation datasets for cross validation (CV). First, hold out 20% for your test dataset. On the remaining 80%, split it into 5 folds. Make sure to seed your random number generator and/or store the indices to ensure your datasets remain consistent through runs.

```
admits <- read.csv("gradAdmit.csv")
```

```
set.seed(314159)
n = nrow(admits) # number of samples
# hold out 20% for testing
sample = sample.int(n = n, size = floor(.2*n), replace = F)
train = admits[-sample,]; test = admits[sample,]
```

```
library(caret)
```

```
nfolds = 5
folds = createFolds(1:n, k=nfolds)
```

Problem 1b

Train a number of SVM models (using different hyperparameters) on the training set for each CV fold. For each run, report the accuracy on both the training and validation datasets, averaged over the folds. Use the same split as Problem 1a. Try using various kernel functions, such as linear, polynomial, radial basis (or Gaussian), etc. Also, try to tune their respective hyperparameters (degree, gamma, and coef0), and the value for cost (or C), based on the validation accuracy. For a full list of the SVM arguments, visit https://www.rdocumentation.org/packages/e1071/versions/1.7-2/topics/svm. Which kernel(s) perform better than others for this problem? Which hypereparameter values are optimal? Which model performed best on the validation dataset? Do NOT use parameter fitting functions from other R packages.

```
library(e1071)
```

```
tuneSVM <- function(kernel,cost,gamma,coef0,degree){

  trainaccyarr = c(0,0,0,0,0)
```

```
valaccarr = c(0,0,0,0,0)

for (i in 1:nfolds){
    train = admits[-folds[[i]],]
    validation = admits[folds[[i]],]
    if(missing(gamma)){ #linear
      smodel <- svm(factor(admit)~.,kernel=kernel,cost=cost, data=train)
    }
    else if(missing(coef0)){ #radial
      smodel <- svm(factor(admit)~.,kernel=kernel,cost=cost,gamma=gamma, data=train)
    }
    else if(missing(degree)){ #sigmoid
      smodel <- svm(factor(admit)~.,kernel=kernel,cost=cost,gamma=gamma, data=train)
    }
    else{ #polynomial
      smodel <- svm(factor(admit)~.,kernel=kernel,cost=cost,gamma=gamma,degree=degree,coef0=coef0,
          data=train)
    }
    trainpred = predict(smodel,newdata=train)
    traintest = confusionMatrix(factor(train$admit), trainpred)
    trainacc = traintest$overall['Accuracy']
    valpredict = predict(smodel,newdata=validation)
    valtest = confusionMatrix(factor(validation$admit), valpredict)
    valacc = valtest$overall['Accuracy']

    trainaccyarr[i] <- trainacc
    valaccarr[i] <- valacc


}

meantrain <- mean(trainaccyarr)
meanval <- mean(valaccarr)

if(missing(gamma)){ #linear
    infodf <- data.frame(

        kernel = kernel,

        cost = cost,

        gamma = "NULL",

        coef0 = "NULL",

        degree = "NULL",

        trainavg = meantrain,

        valavg = meanval

    )
}
```

```
else if(missing(coef0)){ #radial
    infodf <- data.frame(

        kernel = kernel,

        cost = cost,

        gamma = gamma,

        coef0 = "NULL",

        degree = "NULL",

        trainavg = meantrain,

        valavg = meanval

    )
}
else if(missing(degree)){ #sigmoid
    infodf <- data.frame(

        kernel = kernel,

        cost = cost,

        gamma = gamma,

        coef0 = coef0,

        degree = "NULL",

        trainavg = meantrain,

        valavg = meanval

    )
}
else{ #polynomial
    infodf <- data.frame(

        kernel = kernel,

        cost = cost,

        gamma = gamma,

        coef0 = coef0,

        degree = degree,

        trainavg = meantrain,
```

```
          valavg = meanval
      )
  }

  return(infodf)


}
```

```
mdf <- data.frame(

          kernel = character(),

          cost = double(),

          gamma = double(),

          coef0 = double(),

          degree = integer(),

          trainavg = double(),

          valavg = double())
```

```
gammas = c( 0.1, 0.333, 0.5, 0.8, 1)
coef0s = c(0, 0.01, 0.1, 1)
costs = c(0.001, 0.01, 0.1, 1 )
degrees = c(3,4,5)
```

```
for(i in 1:length(costs)){

  ldf = tuneSVM('linear',cost = costs[i])
  mdf <- rbind(mdf, ldf)

}
```

```
for(i in 1:length(costs)){

  for(j in 1:length(gammas)){

    rdf = tuneSVM('radial',cost = costs[i],gamma = gammas[j])
    mdf <- rbind(mdf, rdf)

  }

}
```

```
for(i in 1:length(costs)){

  for(j in 1:length(gammas)){
```

```
    for(k in 1:length(coef0s)){

      sdf = tuneSVM('sigmoid',cost = costs[i],gamma = gammas[j], coef0 = coef0s[k])
      mdf <- rbind(mdf, sdf)

    }

  }

}
```

```
for(i in 1:length(costs)){

  for(j in 1:length(gammas)){

    for(k in 1:length(coef0s)){

      for(l in 1:length(degrees)){

        pdf = tuneSVM('polynomial',cost = costs[i],gamma = gammas[j], coef0 = coef0s[k],degree = degree
        mdf <- rbind(mdf, pdf)

      }

    }

  }

}
```

Results:

```
mdf
```

```
##         kernel  cost gamma coef0 degree trainavg valavg
## 1       linear 0.001  NULL  NULL   NULL 0.682500 0.6825
## 2       linear 0.010  NULL  NULL   NULL 0.682500 0.6825
## 3       linear 0.100  NULL  NULL   NULL 0.682500 0.6825
## 4       linear 1.000  NULL  NULL   NULL 0.682500 0.6825
## 5       radial 0.001   0.1  NULL   NULL 0.682500 0.6825
## 6       radial 0.001 0.333  NULL   NULL 0.682500 0.6825
## 7       radial 0.001   0.5  NULL   NULL 0.682500 0.6825
## 8       radial 0.001   0.8  NULL   NULL 0.682500 0.6825
## 9       radial 0.001     1  NULL   NULL 0.682500 0.6825
## 10      radial 0.010   0.1  NULL   NULL 0.682500 0.6825
## 11      radial 0.010 0.333  NULL   NULL 0.682500 0.6825
## 12      radial 0.010   0.5  NULL   NULL 0.682500 0.6825
## 13      radial 0.010   0.8  NULL   NULL 0.682500 0.6825
## 14      radial 0.010     1  NULL   NULL 0.682500 0.6825
## 15      radial 0.100   0.1  NULL   NULL 0.682500 0.6825
## 16      radial 0.100 0.333  NULL   NULL 0.682500 0.6825
```

```
## 17      radial 0.100   0.5  NULL   NULL 0.682500 0.6825
## 18      radial 0.100   0.8  NULL   NULL 0.682500 0.6825
## 19      radial 0.100     1  NULL   NULL 0.682500 0.6825
## 20      radial 1.000   0.1  NULL   NULL 0.704375 0.6900
## 21      radial 1.000 0.333  NULL   NULL 0.724375 0.7100
## 22      radial 1.000   0.5  NULL   NULL 0.729375 0.7175
## 23      radial 1.000   0.8  NULL   NULL 0.743750 0.7125
## 24      radial 1.000     1  NULL   NULL 0.753750 0.7075
## 25     sigmoid 0.001   0.1     0   NULL 0.682500 0.6825
## 26     sigmoid 0.001   0.1  0.01   NULL 0.682500 0.6825
## 27     sigmoid 0.001   0.1   0.1   NULL 0.682500 0.6825
## 28     sigmoid 0.001   0.1     1   NULL 0.682500 0.6825
## 29     sigmoid 0.001 0.333     0   NULL 0.682500 0.6825
## 30     sigmoid 0.001 0.333  0.01   NULL 0.682500 0.6825
## 31     sigmoid 0.001 0.333   0.1   NULL 0.682500 0.6825
## 32     sigmoid 0.001 0.333     1   NULL 0.682500 0.6825
## 33     sigmoid 0.001   0.5     0   NULL 0.682500 0.6825
## 34     sigmoid 0.001   0.5  0.01   NULL 0.682500 0.6825
## 35     sigmoid 0.001   0.5   0.1   NULL 0.682500 0.6825
## 36     sigmoid 0.001   0.5     1   NULL 0.682500 0.6825
## 37     sigmoid 0.001   0.8     0   NULL 0.682500 0.6825
## 38     sigmoid 0.001   0.8  0.01   NULL 0.682500 0.6825
## 39     sigmoid 0.001   0.8   0.1   NULL 0.682500 0.6825
## 40     sigmoid 0.001   0.8     1   NULL 0.682500 0.6825
## 41     sigmoid 0.001     1     0   NULL 0.682500 0.6825
## 42     sigmoid 0.001     1  0.01   NULL 0.682500 0.6825
## 43     sigmoid 0.001     1   0.1   NULL 0.682500 0.6825
## 44     sigmoid 0.001     1     1   NULL 0.682500 0.6825
## 45     sigmoid 0.010   0.1     0   NULL 0.682500 0.6825
## 46     sigmoid 0.010   0.1  0.01   NULL 0.682500 0.6825
## 47     sigmoid 0.010   0.1   0.1   NULL 0.682500 0.6825
## 48     sigmoid 0.010   0.1     1   NULL 0.682500 0.6825
## 49     sigmoid 0.010 0.333     0   NULL 0.682500 0.6825
## 50     sigmoid 0.010 0.333  0.01   NULL 0.682500 0.6825
## 51     sigmoid 0.010 0.333   0.1   NULL 0.682500 0.6825
## 52     sigmoid 0.010 0.333     1   NULL 0.682500 0.6825
## 53     sigmoid 0.010   0.5     0   NULL 0.682500 0.6825
## 54     sigmoid 0.010   0.5  0.01   NULL 0.682500 0.6825
## 55     sigmoid 0.010   0.5   0.1   NULL 0.682500 0.6825
## 56     sigmoid 0.010   0.5     1   NULL 0.682500 0.6825
## 57     sigmoid 0.010   0.8     0   NULL 0.682500 0.6825
## 58     sigmoid 0.010   0.8  0.01   NULL 0.682500 0.6825
## 59     sigmoid 0.010   0.8   0.1   NULL 0.682500 0.6825
## 60     sigmoid 0.010   0.8     1   NULL 0.682500 0.6825
## 61     sigmoid 0.010     1     0   NULL 0.682500 0.6825
## 62     sigmoid 0.010     1  0.01   NULL 0.682500 0.6825
## 63     sigmoid 0.010     1   0.1   NULL 0.682500 0.6825
## 64     sigmoid 0.010     1     1   NULL 0.682500 0.6825
## 65     sigmoid 0.100   0.1     0   NULL 0.682500 0.6825
## 66     sigmoid 0.100   0.1  0.01   NULL 0.682500 0.6825
## 67     sigmoid 0.100   0.1   0.1   NULL 0.682500 0.6825
## 68     sigmoid 0.100   0.1     1   NULL 0.682500 0.6825
## 69     sigmoid 0.100 0.333     0   NULL 0.682500 0.6825
## 70     sigmoid 0.100 0.333  0.01   NULL 0.682500 0.6825
```

```
## 71     sigmoid 0.100 0.333   0.1   NULL 0.682500 0.6825
## 72     sigmoid 0.100 0.333     1   NULL 0.682500 0.6825
## 73     sigmoid 0.100   0.5     0   NULL 0.678125 0.6750
## 74     sigmoid 0.100   0.5  0.01   NULL 0.678125 0.6750
## 75     sigmoid 0.100   0.5   0.1   NULL 0.678125 0.6750
## 76     sigmoid 0.100   0.5     1   NULL 0.678125 0.6750
## 77     sigmoid 0.100   0.8     0   NULL 0.661875 0.6700
## 78     sigmoid 0.100   0.8  0.01   NULL 0.661875 0.6700
## 79     sigmoid 0.100   0.8   0.1   NULL 0.661875 0.6700
## 80     sigmoid 0.100   0.8     1   NULL 0.661875 0.6700
## 81     sigmoid 0.100     1     0   NULL 0.650625 0.6475
## 82     sigmoid 0.100     1  0.01   NULL 0.650625 0.6475
## 83     sigmoid 0.100     1   0.1   NULL 0.650625 0.6475
## 84     sigmoid 0.100     1     1   NULL 0.650625 0.6475
## 85     sigmoid 1.000   0.1     0   NULL 0.682500 0.6825
## 86     sigmoid 1.000   0.1  0.01   NULL 0.682500 0.6825
## 87     sigmoid 1.000   0.1   0.1   NULL 0.682500 0.6825
## 88     sigmoid 1.000   0.1     1   NULL 0.682500 0.6825
## 89     sigmoid 1.000 0.333     0   NULL 0.616875 0.6425
## 90     sigmoid 1.000 0.333  0.01   NULL 0.616875 0.6425
## 91     sigmoid 1.000 0.333   0.1   NULL 0.616875 0.6425
## 92     sigmoid 1.000 0.333     1   NULL 0.616875 0.6425
## 93     sigmoid 1.000   0.5     0   NULL 0.595625 0.6250
## 94     sigmoid 1.000   0.5  0.01   NULL 0.595625 0.6250
## 95     sigmoid 1.000   0.5   0.1   NULL 0.595625 0.6250
## 96     sigmoid 1.000   0.5     1   NULL 0.595625 0.6250
## 97     sigmoid 1.000   0.8     0   NULL 0.588750 0.6200
## 98     sigmoid 1.000   0.8  0.01   NULL 0.588750 0.6200
## 99     sigmoid 1.000   0.8   0.1   NULL 0.588750 0.6200
## 100    sigmoid 1.000   0.8     1   NULL 0.588750 0.6200
## 101    sigmoid 1.000     1     0   NULL 0.575625 0.5925
## 102    sigmoid 1.000     1  0.01   NULL 0.575625 0.5925
## 103    sigmoid 1.000     1   0.1   NULL 0.575625 0.5925
## 104    sigmoid 1.000     1     1   NULL 0.575625 0.5925
## 105 polynomial 0.001   0.1     0      3 0.682500 0.6825
## 106 polynomial 0.001   0.1     0      4 0.682500 0.6825
## 107 polynomial 0.001   0.1     0      5 0.682500 0.6825
## 108 polynomial 0.001   0.1  0.01      3 0.682500 0.6825
## 109 polynomial 0.001   0.1  0.01      4 0.682500 0.6825
## 110 polynomial 0.001   0.1  0.01      5 0.682500 0.6825
## 111 polynomial 0.001   0.1   0.1      3 0.682500 0.6825
## 112 polynomial 0.001   0.1   0.1      4 0.682500 0.6825
## 113 polynomial 0.001   0.1   0.1      5 0.682500 0.6825
## 114 polynomial 0.001   0.1     1      3 0.682500 0.6825
## 115 polynomial 0.001   0.1     1      4 0.682500 0.6825
## 116 polynomial 0.001   0.1     1      5 0.682500 0.6825
## 117 polynomial 0.001 0.333     0      3 0.682500 0.6825
## 118 polynomial 0.001 0.333     0      4 0.682500 0.6825
## 119 polynomial 0.001 0.333     0      5 0.682500 0.6825
## 120 polynomial 0.001 0.333  0.01      3 0.682500 0.6825
## 121 polynomial 0.001 0.333  0.01      4 0.682500 0.6825
## 122 polynomial 0.001 0.333  0.01      5 0.682500 0.6825
## 123 polynomial 0.001 0.333   0.1      3 0.682500 0.6825
## 124 polynomial 0.001 0.333   0.1      4 0.682500 0.6825
```

```
## 125 polynomial 0.001 0.333   0.1      5 0.682500 0.6825
## 126 polynomial 0.001 0.333     1      3 0.682500 0.6825
## 127 polynomial 0.001 0.333     1      4 0.682500 0.6825
## 128 polynomial 0.001 0.333     1      5 0.689375 0.6850
## 129 polynomial 0.001   0.5     0      3 0.682500 0.6825
## 130 polynomial 0.001   0.5     0      4 0.682500 0.6825
## 131 polynomial 0.001   0.5     0      5 0.687500 0.6850
## 132 polynomial 0.001   0.5  0.01      3 0.682500 0.6825
## 133 polynomial 0.001   0.5  0.01      4 0.682500 0.6825
## 134 polynomial 0.001   0.5  0.01      5 0.688750 0.6850
## 135 polynomial 0.001   0.5   0.1      3 0.682500 0.6825
## 136 polynomial 0.001   0.5   0.1      4 0.682500 0.6825
## 137 polynomial 0.001   0.5   0.1      5 0.692500 0.6875
## 138 polynomial 0.001   0.5     1      3 0.682500 0.6825
## 139 polynomial 0.001   0.5     1      4 0.689375 0.6850
## 140 polynomial 0.001   0.5     1      5 0.705625 0.6950
## 141 polynomial 0.001   0.8     0      3 0.682500 0.6825
## 142 polynomial 0.001   0.8     0      4 0.682500 0.6825
## 143 polynomial 0.001   0.8     0      5 0.706875 0.6850
## 144 polynomial 0.001   0.8  0.01      3 0.682500 0.6825
## 145 polynomial 0.001   0.8  0.01      4 0.682500 0.6825
## 146 polynomial 0.001   0.8  0.01      5 0.706875 0.6875
## 147 polynomial 0.001   0.8   0.1      3 0.682500 0.6825
## 148 polynomial 0.001   0.8   0.1      4 0.683750 0.6825
## 149 polynomial 0.001   0.8   0.1      5 0.709375 0.6875
## 150 polynomial 0.001   0.8     1      3 0.685000 0.6850
## 151 polynomial 0.001   0.8     1      4 0.705625 0.6950
## 152 polynomial 0.001   0.8     1      5 0.721875 0.7025
## 153 polynomial 0.001     1     0      3 0.683125 0.6825
## 154 polynomial 0.001     1     0      4 0.683125 0.6825
## 155 polynomial 0.001     1     0      5 0.713125 0.6900
## 156 polynomial 0.001     1  0.01      3 0.683125 0.6825
## 157 polynomial 0.001     1  0.01      4 0.683125 0.6825
## 158 polynomial 0.001     1  0.01      5 0.713750 0.6875
## 159 polynomial 0.001     1   0.1      3 0.685000 0.6825
## 160 polynomial 0.001     1   0.1      4 0.697500 0.6825
## 161 polynomial 0.001     1   0.1      5 0.718750 0.6925
## 162 polynomial 0.001     1     1      3 0.693125 0.6850
## 163 polynomial 0.001     1     1      4 0.710625 0.7000
## 164 polynomial 0.001     1     1      5 0.731875 0.7025
## 165 polynomial 0.010   0.1     0      3 0.682500 0.6825
## 166 polynomial 0.010   0.1     0      4 0.682500 0.6825
## 167 polynomial 0.010   0.1     0      5 0.682500 0.6825
## 168 polynomial 0.010   0.1  0.01      3 0.682500 0.6825
## 169 polynomial 0.010   0.1  0.01      4 0.682500 0.6825
## 170 polynomial 0.010   0.1  0.01      5 0.682500 0.6825
## 171 polynomial 0.010   0.1   0.1      3 0.682500 0.6825
## 172 polynomial 0.010   0.1   0.1      4 0.682500 0.6825
## 173 polynomial 0.010   0.1   0.1      5 0.682500 0.6825
## 174 polynomial 0.010   0.1     1      3 0.682500 0.6825
## 175 polynomial 0.010   0.1     1      4 0.682500 0.6825
## 176 polynomial 0.010   0.1     1      5 0.682500 0.6825
## 177 polynomial 0.010 0.333     0      3 0.682500 0.6825
## 178 polynomial 0.010 0.333     0      4 0.682500 0.6825
```

```
## 179 polynomial 0.010 0.333     0      5 0.691875 0.6875
## 180 polynomial 0.010 0.333  0.01      3 0.682500 0.6825
## 181 polynomial 0.010 0.333  0.01      4 0.682500 0.6825
## 182 polynomial 0.010 0.333  0.01      5 0.692500 0.6875
## 183 polynomial 0.010 0.333   0.1      3 0.682500 0.6825
## 184 polynomial 0.010 0.333   0.1      4 0.682500 0.6825
## 185 polynomial 0.010 0.333   0.1      5 0.698125 0.6875
## 186 polynomial 0.010 0.333     1      3 0.686875 0.6825
## 187 polynomial 0.010 0.333     1      4 0.705625 0.6900
## 188 polynomial 0.010 0.333     1      5 0.715625 0.7000
## 189 polynomial 0.010   0.5     0      3 0.684375 0.6825
## 190 polynomial 0.010   0.5     0      4 0.682500 0.6825
## 191 polynomial 0.010   0.5     0      5 0.706875 0.6850
## 192 polynomial 0.010   0.5  0.01      3 0.685000 0.6825
## 193 polynomial 0.010   0.5  0.01      4 0.683125 0.6825
## 194 polynomial 0.010   0.5  0.01      5 0.707500 0.6875
## 195 polynomial 0.010   0.5   0.1      3 0.686250 0.6825
## 196 polynomial 0.010   0.5   0.1      4 0.697500 0.6825
## 197 polynomial 0.010   0.5   0.1      5 0.710000 0.6950
## 198 polynomial 0.010   0.5     1      3 0.703125 0.6875
## 199 polynomial 0.010   0.5     1      4 0.715000 0.7000
## 200 polynomial 0.010   0.5     1      5 0.729375 0.7100
## 201 polynomial 0.010   0.8     0      3 0.694375 0.6850
## 202 polynomial 0.010   0.8     0      4 0.683750 0.6775
## 203 polynomial 0.010   0.8     0      5 0.728750 0.6925
## 204 polynomial 0.010   0.8  0.01      3 0.694375 0.6850
## 205 polynomial 0.010   0.8  0.01      4 0.691250 0.6750
## 206 polynomial 0.010   0.8  0.01      5 0.732500 0.6975
## 207 polynomial 0.010   0.8   0.1      3 0.701250 0.6825
## 208 polynomial 0.010   0.8   0.1      4 0.709375 0.7000
## 209 polynomial 0.010   0.8   0.1      5 0.739375 0.7000
## 210 polynomial 0.010   0.8     1      3 0.709375 0.7000
## 211 polynomial 0.010   0.8     1      4 0.723750 0.7000
## 212 polynomial 0.010   0.8     1      5 0.752500 0.6925
## 213 polynomial 0.010     1     0      3 0.698125 0.6775
## 214 polynomial 0.010     1     0      4 0.683750 0.6725
## 215 polynomial 0.010     1     0      5 0.746250 0.7050
## 216 polynomial 0.010     1  0.01      3 0.698125 0.6775
## 217 polynomial 0.010     1  0.01      4 0.700625 0.6925
## 218 polynomial 0.010     1  0.01      5 0.748125 0.7075
## 219 polynomial 0.010     1   0.1      3 0.703125 0.6900
## 220 polynomial 0.010     1   0.1      4 0.718125 0.7100
## 221 polynomial 0.010     1   0.1      5 0.751250 0.6950
## 222 polynomial 0.010     1     1      3 0.713750 0.7050
## 223 polynomial 0.010     1     1      4 0.722500 0.7025
## 224 polynomial 0.010     1     1      5 0.760000 0.6900
## 225 polynomial 0.100   0.1     0      3 0.682500 0.6825
## 226 polynomial 0.100   0.1     0      4 0.682500 0.6825
## 227 polynomial 0.100   0.1     0      5 0.682500 0.6825
## 228 polynomial 0.100   0.1  0.01      3 0.682500 0.6825
## 229 polynomial 0.100   0.1  0.01      4 0.682500 0.6825
## 230 polynomial 0.100   0.1  0.01      5 0.682500 0.6825
## 231 polynomial 0.100   0.1   0.1      3 0.682500 0.6825
## 232 polynomial 0.100   0.1   0.1      4 0.682500 0.6825
```

```
## 233 polynomial 0.100   0.1   0.1      5 0.682500 0.6825
## 234 polynomial 0.100   0.1     1      3 0.683750 0.6825
## 235 polynomial 0.100   0.1     1      4 0.700000 0.6850
## 236 polynomial 0.100   0.1     1      5 0.705000 0.6975
## 237 polynomial 0.100 0.333     0      3 0.692500 0.6825
## 238 polynomial 0.100 0.333     0      4 0.683125 0.6825
## 239 polynomial 0.100 0.333     0      5 0.706875 0.6875
## 240 polynomial 0.100 0.333  0.01      3 0.692500 0.6850
## 241 polynomial 0.100 0.333  0.01      4 0.684375 0.6825
## 242 polynomial 0.100 0.333  0.01      5 0.706875 0.6875
## 243 polynomial 0.100 0.333   0.1      3 0.702500 0.6875
## 244 polynomial 0.100 0.333   0.1      4 0.705625 0.6950
## 245 polynomial 0.100 0.333   0.1      5 0.714375 0.6900
## 246 polynomial 0.100 0.333     1      3 0.710625 0.7050
## 247 polynomial 0.100 0.333     1      4 0.720000 0.7025
## 248 polynomial 0.100 0.333     1      5 0.731875 0.7050
## 249 polynomial 0.100   0.5     0      3 0.700000 0.6775
## 250 polynomial 0.100   0.5     0      4 0.685625 0.6750
## 251 polynomial 0.100   0.5     0      5 0.727500 0.6900
## 252 polynomial 0.100   0.5  0.01      3 0.701250 0.6825
## 253 polynomial 0.100   0.5  0.01      4 0.703750 0.6950
## 254 polynomial 0.100   0.5  0.01      5 0.730625 0.7025
## 255 polynomial 0.100   0.5   0.1      3 0.708125 0.6925
## 256 polynomial 0.100   0.5   0.1      4 0.718750 0.7075
## 257 polynomial 0.100   0.5   0.1      5 0.739375 0.6975
## 258 polynomial 0.100   0.5     1      3 0.715625 0.7075
## 259 polynomial 0.100   0.5     1      4 0.723125 0.7000
## 260 polynomial 0.100   0.5     1      5 0.751875 0.6950
## 261 polynomial 0.100   0.8     0      3 0.704375 0.6875
## 262 polynomial 0.100   0.8     0      4 0.683750 0.6725
## 263 polynomial 0.100   0.8     0      5 0.748750 0.7200
## 264 polynomial 0.100   0.8  0.01      3 0.705625 0.6900
## 265 polynomial 0.100   0.8  0.01      4 0.715625 0.7050
## 266 polynomial 0.100   0.8  0.01      5 0.752500 0.7000
## 267 polynomial 0.100   0.8   0.1      3 0.715000 0.7050
## 268 polynomial 0.100   0.8   0.1      4 0.724375 0.7000
## 269 polynomial 0.100   0.8   0.1      5 0.758750 0.6850
## 270 polynomial 0.100   0.8     1      3 0.715625 0.7075
## 271 polynomial 0.100   0.8     1      4 0.726250 0.6975
## 272 polynomial 0.100   0.8     1      5 0.772500 0.6775
## 273 polynomial 0.100     1     0      3 0.704375 0.6875
## 274 polynomial 0.100     1     0      4 0.683750 0.6725
## 275 polynomial 0.100     1     0      5 0.750000 0.7175
## 276 polynomial 0.100     1  0.01      3 0.707500 0.6900
## 277 polynomial 0.100     1  0.01      4 0.720625 0.7125
## 278 polynomial 0.100     1  0.01      5 0.753750 0.6875
## 279 polynomial 0.100     1   0.1      3 0.716250 0.7050
## 280 polynomial 0.100     1   0.1      4 0.724375 0.6975
## 281 polynomial 0.100     1   0.1      5 0.763125 0.6750
## 282 polynomial 0.100     1     1      3 0.716875 0.7075
## 283 polynomial 0.100     1     1      4 0.726250 0.7000
## 284 polynomial 0.100     1     1      5 0.778125 0.6700
## 285 polynomial 1.000   0.1     0      3 0.683125 0.6825
## 286 polynomial 1.000   0.1     0      4 0.682500 0.6825
```

```
## 287 polynomial 1.000   0.1     0      5 0.682500 0.6825
## 288 polynomial 1.000   0.1  0.01      3 0.685000 0.6825
## 289 polynomial 1.000   0.1  0.01      4 0.682500 0.6825
## 290 polynomial 1.000   0.1  0.01      5 0.682500 0.6825
## 291 polynomial 1.000   0.1   0.1      3 0.693125 0.6850
## 292 polynomial 1.000   0.1   0.1      4 0.686250 0.6850
## 293 polynomial 1.000   0.1   0.1      5 0.686250 0.6825
## 294 polynomial 1.000   0.1     1      3 0.710000 0.6950
## 295 polynomial 1.000   0.1     1      4 0.715625 0.7050
## 296 polynomial 1.000   0.1     1      5 0.722500 0.7075
## 297 polynomial 1.000 0.333     0      3 0.703750 0.6875
## 298 polynomial 1.000 0.333     0      4 0.683750 0.6725
## 299 polynomial 1.000 0.333     0      5 0.731250 0.6925
## 300 polynomial 1.000 0.333  0.01      3 0.708125 0.6900
## 301 polynomial 1.000 0.333  0.01      4 0.710000 0.7000
## 302 polynomial 1.000 0.333  0.01      5 0.738750 0.7050
## 303 polynomial 1.000 0.333   0.1      3 0.716250 0.7050
## 304 polynomial 1.000 0.333   0.1      4 0.723125 0.7025
## 305 polynomial 1.000 0.333   0.1      5 0.743125 0.7025
## 306 polynomial 1.000 0.333     1      3 0.717500 0.7075
## 307 polynomial 1.000 0.333     1      4 0.723750 0.7000
## 308 polynomial 1.000 0.333     1      5 0.762500 0.6800
## 309 polynomial 1.000   0.5     0      3 0.704375 0.6875
## 310 polynomial 1.000   0.5     0      4 0.683750 0.6725
## 311 polynomial 1.000   0.5     0      5 0.750000 0.7175
## 312 polynomial 1.000   0.5  0.01      3 0.706250 0.6950
## 313 polynomial 1.000   0.5  0.01      4 0.721250 0.7050
## 314 polynomial 1.000   0.5  0.01      5 0.751250 0.6900
## 315 polynomial 1.000   0.5   0.1      3 0.717500 0.7050
## 316 polynomial 1.000   0.5   0.1      4 0.723750 0.7000
## 317 polynomial 1.000   0.5   0.1      5 0.760625 0.6825
## 318 polynomial 1.000   0.5     1      3 0.718125 0.7100
## 319 polynomial 1.000   0.5     1      4 0.726250 0.7000
## 320 polynomial 1.000   0.5     1      5 0.778125 0.6725
## 321 polynomial 1.000   0.8     0      3 0.705000 0.6875
## 322 polynomial 1.000   0.8     0      4 0.683750 0.6725
## 323 polynomial 1.000   0.8     0      5 0.751250 0.7200
## 324 polynomial 1.000   0.8  0.01      3 0.715000 0.7025
## 325 polynomial 1.000   0.8  0.01      4 0.724375 0.7025
## 326 polynomial 1.000   0.8  0.01      5 0.757500 0.6800
## 327 polynomial 1.000   0.8   0.1      3 0.717500 0.7075
## 328 polynomial 1.000   0.8   0.1      4 0.724375 0.6975
## 329 polynomial 1.000   0.8   0.1      5 0.768125 0.6800
## 330 polynomial 1.000   0.8     1      3 0.718750 0.7075
## 331 polynomial 1.000   0.8     1      4 0.729375 0.7000
## 332 polynomial 1.000   0.8     1      5 0.785625 0.6650
## 333 polynomial 1.000     1     0      3 0.705625 0.6875
## 334 polynomial 1.000     1     0      4 0.683750 0.6725
## 335 polynomial 1.000     1     0      5 0.750625 0.7200
## 336 polynomial 1.000     1  0.01      3 0.715625 0.7025
## 337 polynomial 1.000     1  0.01      4 0.723750 0.7000
## 338 polynomial 1.000     1  0.01      5 0.760625 0.6700
## 339 polynomial 1.000     1   0.1      3 0.716250 0.7075
## 340 polynomial 1.000     1   0.1      4 0.723750 0.6975
```

```
## 341 polynomial 1.000     1   0.1      5 0.770000 0.6825
## 342 polynomial 1.000     1     1      3 0.718750 0.7075
## 343 polynomial 1.000     1     1      4 0.730000 0.7000
## 344 polynomial 1.000     1     1      5 0.788750 0.6700
```

```
library(data.table)
mdt <- data.table(mdf)
```

Find out which hyperparameters performed the best

```
mdt[,max(valavg),by=kernel]
```

```
##        kernel     V1
## 1:     linear 0.6825
## 2:     radial 0.7175
## 3:    sigmoid 0.6825
## 4: polynomial 0.7200
```

```
mdt[mdt[, .I[valavg == max(valavg)]]]
```

```
##        kernel cost gamma coef0 degree trainavg valavg
## 1: polynomial  0.1   0.8     0      5 0.748750   0.72
## 2: polynomial  1.0   0.8     0      5 0.751250   0.72
## 3: polynomial  1.0     1     0      5 0.750625   0.72
```

Radial and polynomial kernels performed better than linear and sigmoid. The highest average accuracies on the validation folds were from the polynomial kernel. Three sets of hyperparameters tied in their average validation accuracy, shown above. All had degree at 5 and coef0 at 0, while cost and gamma differed.

Problem 1(c)

For your best model from Problem 1b, retrain it on the full training set (the 80% that was used for training and validation) and compute the accuracy on the test dataset (the 20% that until this point was untouched)

I arbitrarily chose the model with a polynomial kernel, a cost of 1, gamma of 1, coef0 of 0, and a degree of 5 out of the three models that performed the best on average for the validation folds.

```
n = nrow(admits)
sample = sample.int(n = n, size = floor(.2*n), replace = F)
train = admits[-sample,]; test = admits[sample,]
```

```
smodel <- svm(factor(admit)~.,kernel='polynomial',cost=1,gamma=1,coef0=0,degree=5, data=train)
testpredict = predict(smodel,newdata=test)
finaltest = confusionMatrix(factor(test$admit), testpredict)
testacc = finaltest$overall['Accuracy']
testacc
```

```
## Accuracy
##      0.7
```

This model predicted accurately 70% of the test set, slightly lower than the average validation accuracy received before.