Deanna Springgay

Ch 13 Hw

13.1

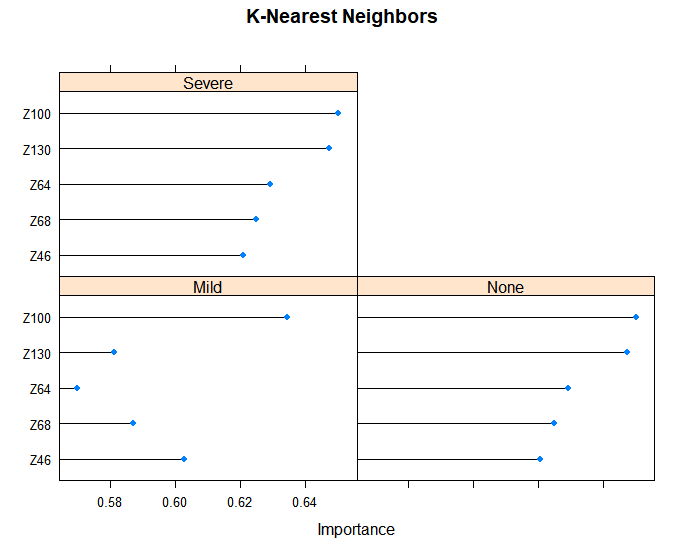
a)

|  |  |  |  |
| --- | --- | --- | --- |
| **Model** | **Tuning Parameter** | **Training Kappa** | **Testing Kappa** |
| Regularized Discriminant Analysis | gamma = 1  lambda = 1 | 3.529610e-02 | 0.1634 |
| Mixture Discriminant Analysis | subclasses = 3 | 0.05477389 | 0.1325 |
| Neural Networks | size = 3  decay = 0 | 0.0449682834 | -0.0554 |
| Support Vector Machines (Radial) | sigma = 0.00286574  C = 16 | 1.589726e-02 | 0 |
| K-Nearest Neighbors | k = 3 | 0.0355894510 | 0.2924 |
| Naïve Bayes | fL = 0  usekernel = TRUE  adjust = TRUE | 0.03526482 | 0.0136 |

K-nearest neighbors has the best predictive ability of the non-linear classification models.

b) The non-linear models appear to perform better on the testing dataset than the linear classification models. The linear models did do better on the training data, but as predictive models the non-linear models would be better on new data. The best model out of the group would be KNN, as it had a training kappa close to the median and the highest testing kappa.

c) For K-nearest neighbors the top five predictors are as follows: Z100, Z130, Z64, Z68, and Z46.



**Model Outputs**

**Regularized Discriminant Analysis**

225 samples

184 predictors

3 classes: 'Mild', 'None', 'Severe'

No pre-processing

Resampling: Bootstrapped (25 reps)

Summary of sample sizes: 225, 225, 225, 225, 225, 225, ...

Resampling results across tuning parameters:

lambda gamma Accuracy Kappa

1 1 0.3707418 3.529610e-02

1 2 0.2025533 -1.961731e-02

1 3 0.2767373 -5.218200e-03

1 4 0.2420699 9.455816e-03

1 5 0.1820371 -1.521001e-02

2 1 0.3378602 -3.250073e-02

2 2 0.2913672 -1.238532e-03

2 3 0.2656037 -2.892103e-03

2 4 0.3164390 -3.148635e-03

2 5 0.2468904 -1.817056e-03

3 1 0.3361571 -4.616737e-02

3 2 0.2738473 1.468682e-03

3 3 0.3024588 3.457777e-03

3 4 0.1954395 3.117338e-03

3 5 0.3112040 6.794838e-03

4 1 0.3557737 -3.572629e-02

4 2 0.2866634 -8.476945e-03

4 3 0.2561993 1.542634e-03

4 4 0.3043105 -2.520141e-03

4 5 0.2331257 -5.491068e-03

5 1 0.3651171 -2.838072e-02

5 2 0.2842211 -9.638377e-03

5 3 0.2228610 9.584877e-04

5 4 0.2286988 -4.321385e-05

5 5 0.2557507 -3.963184e-03

Kappa was used to select the optimal model using the largest value.

The final values used for the model were gamma = 1 and lambda = 1.

Confusion Matrix and Statistics

Reference

Prediction Mild None Severe

Mild 9 3 1

None 5 9 1

Severe 15 9 4

Overall Statistics

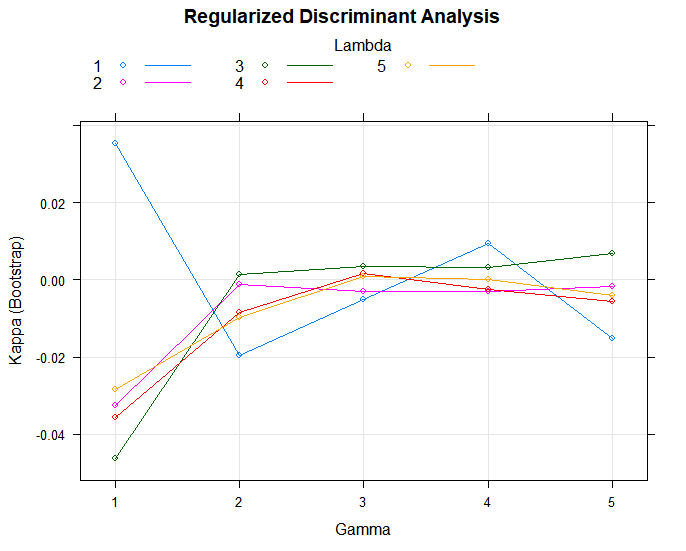
Accuracy : 0.3929

95% CI : (0.265, 0.5325)

No Information Rate : 0.5179

P-Value [Acc > NIR] : 0.9777836

Kappa : 0.1634



**Mixture Discriminant Analysis**

225 samples

184 predictors

3 classes: 'Mild', 'None', 'Severe'

Pre-processing: nearest neighbor imputation (103), centered (103), scaled (103), remove (81)

Resampling: Bootstrapped (25 reps)

Summary of sample sizes: 225, 225, 225, 225, 225, 225, ...

Resampling results across tuning parameters:

subclasses Accuracy Kappa

1 0.4015612 0.03276885

2 0.4042414 0.03300925

3 0.4163232 0.05477389

4 0.4063893 0.04774310

5 0.3836208 0.02465261

Kappa was used to select the optimal model using the largest value.

The final value used for the model was subclasses = 3.

Confusion Matrix and Statistics

Reference

Prediction Mild None Severe

Mild 17 6 3

None 11 9 2

Severe 1 6 1

Overall Statistics

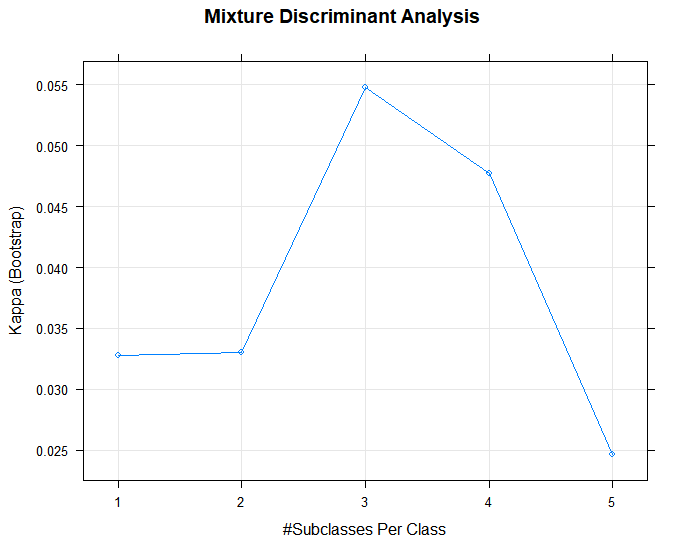
Accuracy : 0.4821

95% CI : (0.3466, 0.6197)

No Information Rate : 0.5179

P-Value [Acc > NIR] : 0.7482

Kappa : 0.1325



**Neural Network**

225 samples

184 predictors

3 classes: 'Mild', 'None', 'Severe'

Pre-processing: centered (103), scaled (103), spatial sign transformation (103), remove (81)

Resampling: Bootstrapped (25 reps)

Summary of sample sizes: 225, 225, 225, 225, 225, 225, ...

Resampling results across tuning parameters:

size decay Accuracy Kappa

1 0.0 0.3998842 0.0189253992

1 0.1 0.4331976 -0.0244385209

1 1.0 0.4788875 0.0062342352

1 2.0 0.4917068 0.0000000000

2 0.0 0.3913583 0.0005663226

2 0.1 0.4314331 0.0058929260

2 1.0 0.4749980 0.0109369407

2 2.0 0.4917068 0.0000000000

3 0.0 0.4314167 0.0449682834

3 0.1 0.4386716 0.0198379778

3 1.0 0.4764894 0.0144213207

3 2.0 0.4917068 0.0000000000

4 0.0 0.4069581 0.0065779676

4 0.1 0.4413163 0.0201787498

4 1.0 0.4769773 0.0157343958

4 2.0 0.4917068 0.0000000000

5 0.0 0.4326719 0.0436599867

5 0.1 0.4409928 0.0221493147

5 1.0 0.4765278 0.0149161838

5 2.0 0.4917068 0.0000000000

6 0.0 0.4381546 0.0339225666

6 0.1 0.4318461 0.0066869529

6 1.0 0.4760150 0.0142718078

6 2.0 0.4917068 0.0000000000

7 0.0 0.4336624 0.0267910111

7 0.1 0.4387746 0.0203264278

7 1.0 0.4765028 0.0151302181

7 2.0 0.4917068 0.0000000000

8 0.0 0.4274128 0.0261944177

8 0.1 0.4425387 0.0221968215

8 1.0 0.4769522 0.0159029177

8 2.0 0.4917068 0.0000000000

9 0.0 0.4331608 0.0284398562

9 0.1 0.4455230 0.0306907576

9 1.0 0.4769242 0.0160739209

9 2.0 0.4917068 0.0000000000

10 0.0 0.4329768 0.0272026089

10 0.1 0.4343217 0.0130136988

10 1.0 0.4764644 0.0150269895

10 2.0 0.4917068 0.0000000000

11 0.0 0.4377164 0.0432389630

11 0.1 0.4369295 0.0152864992

11 1.0 0.4769242 0.0160739209

11 2.0 0.4917068 0.0000000000

12 0.0 0.4390242 0.0369575204

12 0.1 0.4399539 0.0203011856

12 1.0 0.4764114 0.0151904866

12 2.0 0.4917068 0.0000000000

13 0.0 0.4385069 0.0393308166

13 0.1 0.4328418 0.0092086587

13 1.0 0.4764114 0.0151904866

13 2.0 0.4917068 0.0000000000

14 0.0 0.4403095 0.0396739753

14 0.1 0.4339434 0.0113076839

14 1.0 0.4764114 0.0151904866

14 2.0 0.4917068 0.0000000000

15 0.0 0.4303424 0.0302633763

15 0.1 0.4271411 -0.0016545578

15 1.0 0.4764114 0.0151904866

15 2.0 0.4917068 0.0000000000

16 0.0 0.4395065 0.0371242422

16 0.1 0.4389336 0.0190724142

16 1.0 0.4759408 0.0143654928

16 2.0 0.4917068 0.0000000000

17 0.0 0.4265400 0.0191112093

17 0.1 0.4302940 0.0050486413

17 1.0 0.4764536 0.0153970712

17 2.0 0.4917068 0.0000000000

18 0.0 0.4321526 0.0267037406

18 0.1 0.4259409 -0.0025411956

18 1.0 0.4764536 0.0153970712

18 2.0 0.4917068 0.0000000000

19 0.0 0.4340990 0.0180257374

19 0.1 0.4289028 0.0040525020

19 1.0 0.4764536 0.0153970712

19 2.0 0.4917068 0.0000000000

20 0.0 0.4341899 0.0305972621

20 0.1 0.4339767 0.0142240691

20 1.0 0.4764536 0.0153970712

20 2.0 0.4917068 0.0000000000

Kappa was used to select the optimal model using the largest value.

The final values used for the model were size = 3 and decay = 0.

Confusion Matrix and Statistics

Reference

Prediction Mild None Severe

Mild 12 11 4

None 15 9 1

Severe 2 1 1

Overall Statistics

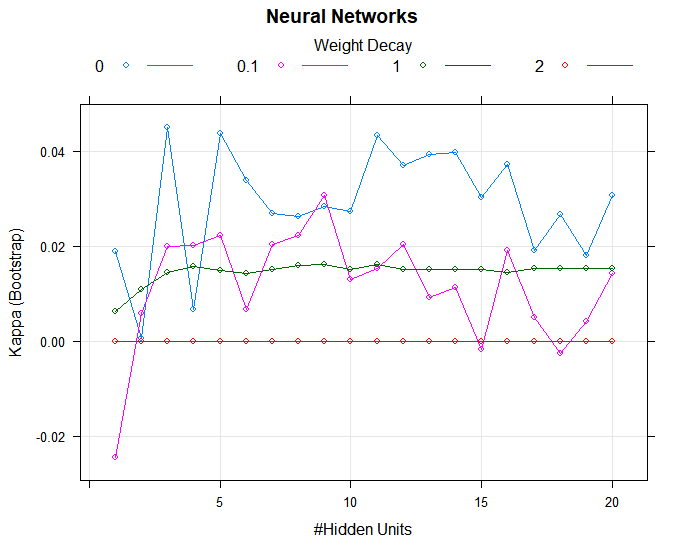
Accuracy : 0.3929

95% CI : (0.265, 0.5325)

No Information Rate : 0.5179

P-Value [Acc > NIR] : 0.9778

Kappa : -0.0554



**Support Vector Machines with Radial Basis Function Kernel**

225 samples

184 predictors

3 classes: 'Mild', 'None', 'Severe'

No pre-processing

Resampling: Bootstrapped (25 reps)

Summary of sample sizes: 225, 225, 225, 225, 225, 225, ...

Resampling results across tuning parameters:

C Accuracy Kappa

0.0625 0.4571952 -3.481604e-03

0.1250 0.4534728 3.906858e-05

0.2500 0.4470309 -1.008623e-02

0.5000 0.4435778 -1.770684e-02

1.0000 0.4431401 -1.346805e-02

2.0000 0.4447095 -1.217018e-02

4.0000 0.4409248 -1.334616e-02

8.0000 0.4510038 2.570743e-03

16.0000 0.4564565 1.589726e-02

32.0000 0.4437872 -6.384103e-03

64.0000 0.4473517 1.795879e-03

Tuning parameter 'sigma' was held constant at a value of 0.00286574

Kappa was used to select the optimal model using the largest value.

The final values used for the model were sigma = 0.00286574 and C = 16.

Confusion Matrix and Statistics

Reference

Prediction Mild None Severe

Mild 29 21 6

None 0 0 0

Severe 0 0 0

Overall Statistics

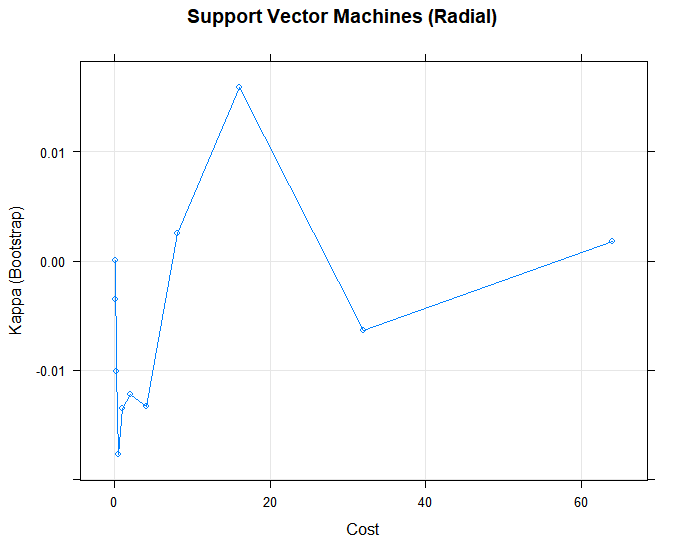
Accuracy : 0.5179

95% CI : (0.3803, 0.6534)

No Information Rate : 0.5179

P-Value [Acc > NIR] : 0.5537

Kappa : 0



**k-Nearest Neighbors**

225 samples

184 predictors

3 classes: 'Mild', 'None', 'Severe'

Pre-processing: centered (103), scaled (103), remove (81)

Resampling: Bootstrapped (25 reps)

Summary of sample sizes: 225, 225, 225, 225, 225, 225, ...

Resampling results across tuning parameters:

k Accuracy Kappa

1 0.4185452 0.0104937944

2 0.4268081 0.0213695518

3 0.4377845 0.0355894510

4 0.4270600 0.0044065506

5 0.4320982 0.0077291435

6 0.4266014 -0.0039427354

7 0.4411511 0.0154321070

8 0.4454609 0.0087360769

9 0.4506702 0.0150773181

10 0.4596744 0.0290057193

11 0.4611993 0.0154106185

12 0.4628086 0.0083258561

13 0.4574838 -0.0027355228

14 0.4654438 0.0081312628

15 0.4684865 0.0091360428

16 0.4740031 0.0124430414

17 0.4844551 0.0289445236

18 0.4823408 0.0232819522

19 0.4807907 0.0170554836

20 0.4820383 0.0139716160

21 0.4730351 -0.0052887662

22 0.4754819 -0.0037379213

23 0.4773053 -0.0034205261

24 0.4783086 0.0008076685

25 0.4781787 -0.0009360249

26 0.4830425 0.0061987810

27 0.4853342 0.0112340237

28 0.4897839 0.0181388834

29 0.4861181 0.0104358099

30 0.4824017 0.0010145016

31 0.4879306 0.0105364666

32 0.4845702 0.0039996532

33 0.4790867 -0.0076842591

34 0.4858130 0.0030296434

35 0.4859844 0.0042006400

36 0.4868228 0.0028896446

37 0.4805159 -0.0084243195

38 0.4834546 -0.0016968599

39 0.4849014 0.0009808176

40 0.4885523 0.0073579607

41 0.4891538 0.0073817182

42 0.4901457 0.0095600455

43 0.4920835 0.0131129536

44 0.4944295 0.0171120255

45 0.4896242 0.0073568923

46 0.4902931 0.0093043377

47 0.4911953 0.0111080038

48 0.4920590 0.0116793885

49 0.4902183 0.0085279344

50 0.4907708 0.0086891300

Kappa was used to select the optimal model using the largest value.

The final value used for the model was k = 3.

Confusion Matrix and Statistics

Reference

Prediction Mild None Severe

Mild 22 9 2

None 6 11 3

Severe 1 1 1

Overall Statistics

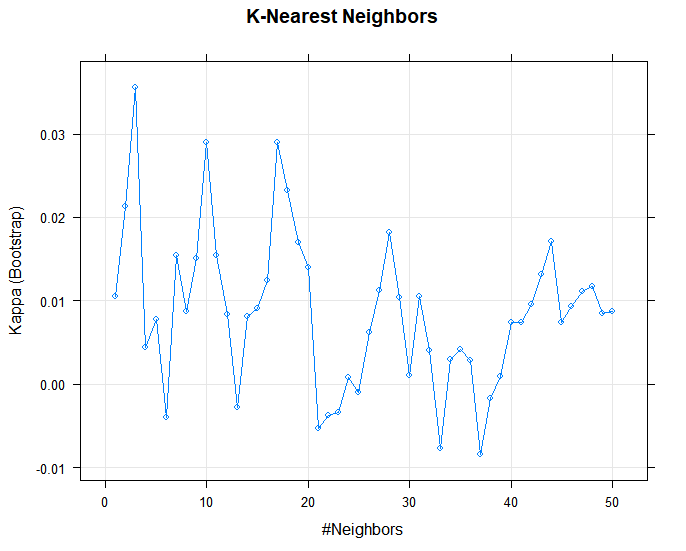
Accuracy : 0.6071

95% CI : (0.4675, 0.735)

No Information Rate : 0.5179

P-Value [Acc > NIR] : 0.1141

Kappa : 0.2924



**Naive Bayes**

225 samples

184 predictors

3 classes: 'Mild', 'None', 'Severe'

Pre-processing: centered (103), scaled (103), remove (81)

Resampling: Bootstrapped (25 reps)

Summary of sample sizes: 225, 225, 225, 225, 225, 225, ...

Resampling results across tuning parameters:

fL Accuracy Kappa

0.0 0.255807 0.03526482

0.1 0.255807 0.03526482

1.0 0.255807 0.03526482

2.0 0.255807 0.03526482

Tuning parameter 'usekernel' was held constant at a value of TRUE

Tuning parameter 'adjust' was held constant at a value of TRUE

Kappa was used to select the optimal model using the largest value.

The final values used for the model were fL = 0, usekernel = TRUE and adjust = TRUE.

Confusion Matrix and Statistics

Reference

Prediction Mild None Severe

Mild 0 4 0

None 6 7 1

Severe 23 10 5

Overall Statistics

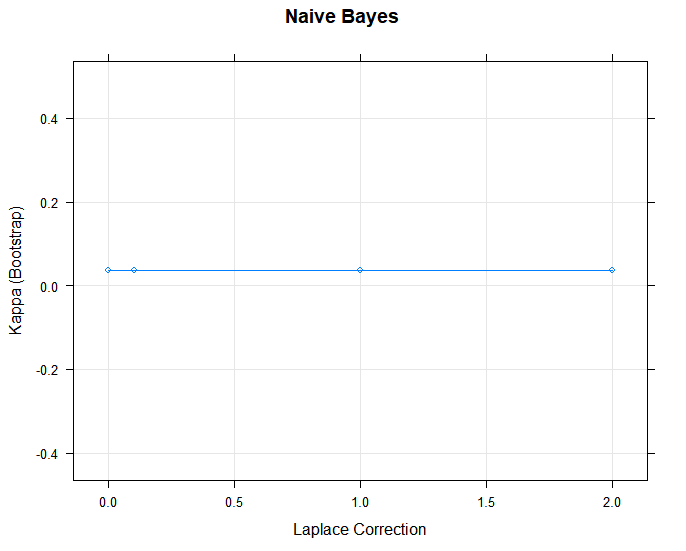
Accuracy : 0.2143

95% CI : (0.1159, 0.3444)

No Information Rate : 0.5179

P-Value [Acc > NIR] : 1

Kappa : 0.0136



**R Code**

library(AppliedPredictiveModeling)

library(caret)

library(glmnet)

library(kernlab)

library(klaR)

library(MASS)

library(tidyverse)

# 13.1 ------------------------------------------------------------------------

data(hepatic)

# creating dataset

full\_dataset <- bio

full\_dataset[, 185] <- injury

# injury is named V185

# splitting data

set.seed(80)

training\_rows <- createDataPartition(full\_dataset[, 185], p = 0.80, list = FALSE)

training\_set <- full\_dataset[training\_rows, ]

testing\_set <- full\_dataset[-training\_rows, ]

# will determine the best model using kappa since there are 3 classes

ctrl <- trainControl(

summaryFunction = defaultSummary,

classProbs = TRUE,

savePredictions = TRUE

)

# Quadratic Discriminant Analysis - n>p, so this model is not applicable

# Regularized Discriminant Analysis

set.seed(80)

rda\_model <- train(

x = training\_set[, 1:184],

y = training\_set$V185,

method = "rda",

metric = "Kappa",

tuneGrid = expand.grid(.lambda = 1:5, .gamma = 1:5),

trControl = ctrl

)

rda\_model

plot(rda\_model, main = "Regularized Discriminant Analysis")

rda\_pred <- predict(rda\_model, newdata = testing\_set[, 1:184])

confusionMatrix(data = rda\_pred, reference = testing\_set$V185)

# Mixture Discriminant Analysis

set.seed(80)

mda\_model <- train(

x = training\_set[, 1:184],

y = training\_set$V185,

method = "mda",

metric = "Kappa",

preProc = c("knnImpute", "nzv"),

tuneGrid = expand.grid(.subclasses = 1:5),

trControl = ctrl

)

mda\_model

plot(mda\_model, main = "Mixture Discriminant Analysis")

mda\_pred <- predict(mda\_model, newdata = testing\_set[, 1:184])

confusionMatrix(data = mda\_pred, reference = testing\_set$V185)

# Neural Networks

nnet\_grid <- expand.grid(.size = 1:20, .decay = c(0, .1, 1, 2))

max\_size <- max(nnet\_grid$.size)

num\_weights <- (max\_size \* (184 + 1) + (max\_size + 1) \* 3) # 184 predictors, 3 classes

set.seed(80)

nnet\_model <- train(

x = training\_set[, 1:184],

y = training\_set$V185,

method = "nnet",

metric = "Kappa",

preProc = c("center", "scale", "spatialSign", "nzv"),

tuneGrid = nnet\_grid,

trace = FALSE,

maxit = 2000,

MaxNWts = num\_weights,

trControl = ctrl

)

nnet\_model

plot(nnet\_model, main = "Neural Networks")

nnet\_pred <- predict(nnet\_model, newdata = testing\_set[, 1:184])

confusionMatrix(data = nnet\_pred, reference = testing\_set$V185)

# Flexible Discriminant Analysis - n>p, so this model is not applicable

# Support Vector Machine (Radial)

sigma\_reduced\_range <- sigest(as.matrix(training\_set[, 1:184]))

svm\_reduced\_grid <- expand.grid(

.sigma = sigma\_reduced\_range[1],

.C = 2^(seq(-4, 6))

)

set.seed(80)

svm\_model <- train(

x = training\_set[, 1:184],

y = training\_set$V185,

method = "svmRadial",

metric = "Kappa",

tuneGrid = svm\_reduced\_grid,

fir = FALSE,

trControl = ctrl

)

svm\_model

plot(svm\_model, main = "Support Vector Machines (Radial)")

svm\_pred <- predict(svm\_model, newdata = testing\_set[, 1:184])

confusionMatrix(data = svm\_pred, reference = testing\_set$V185)

# K-Nearest Neighbors

set.seed(80)

knn\_model <- train(

x = training\_set[, 1:184],

y = training\_set$V185,

method = "knn",

metric = "Kappa",

preProc = c("center", "scale", "nzv"),

tuneGrid = data.frame(.k = 1:50),

trControl = ctrl

)

knn\_model

plot(knn\_model, main = "K-Nearest Neighbors")

knn\_pred <- predict(knn\_model, newdata = testing\_set[, 1:184])

confusionMatrix(data = knn\_pred, reference = testing\_set$V185)

# Naive Bayes

set.seed(80)

nb\_model <- train(

x = training\_set[, 1:184],

y = training\_set$V185,

method = "nb",

metric = "Kappa",

preProc = c("center", "scale", "nzv"),

tuneGrid = data.frame(.fL = c(0, .1, 1, 2), .usekernel = TRUE, .adjust = TRUE),

trControl = ctrl

)

nb\_model

plot(nb\_model, main = "Naive Bayes")

nb\_pred <- predict(nb\_model, newdata = testing\_set[, 1:184])

confusionMatrix(data = nb\_pred, reference = testing\_set$V185)

# variable importance

imp\_vars <- varImp(knn\_model, scale = FALSE)

plot(imp\_vars, top = 5, main = "K-Nearest Neighbors")