> zoo\_df <- read.csv(file.choose())

> zoo1 <- zoo\_df[,-1]

> View(zoo1)

> head(zoo1)

hair feathers eggs milk airborne aquatic predator toothed backbone breathes

1 1 0 0 1 0 0 1 1 1 1

2 1 0 0 1 0 0 0 1 1 1

3 0 0 1 0 0 1 1 1 1 0

4 1 0 0 1 0 0 1 1 1 1

5 1 0 0 1 0 0 1 1 1 1

6 1 0 0 1 0 0 0 1 1 1

venomous fins legs tail domestic catsize type

1 0 0 4 0 0 1 1

2 0 0 4 1 0 1 1

3 0 1 0 1 0 0 4

4 0 0 4 0 0 1 1

5 0 0 4 1 0 1 1

6 0 0 4 1 0 1 1

> summary(zoo1)

hair feathers eggs milk

Min. :0.0000 Min. :0.000 Min. :0.0000 Min. :0.0000

1st Qu.:0.0000 1st Qu.:0.000 1st Qu.:0.0000 1st Qu.:0.0000

Median :0.0000 Median :0.000 Median :1.0000 Median :0.0000

Mean :0.4257 Mean :0.198 Mean :0.5842 Mean :0.4059

3rd Qu.:1.0000 3rd Qu.:0.000 3rd Qu.:1.0000 3rd Qu.:1.0000

Max. :1.0000 Max. :1.000 Max. :1.0000 Max. :1.0000

airborne aquatic predator toothed

Min. :0.0000 Min. :0.0000 Min. :0.0000 Min. :0.000

1st Qu.:0.0000 1st Qu.:0.0000 1st Qu.:0.0000 1st Qu.:0.000

Median :0.0000 Median :0.0000 Median :1.0000 Median :1.000

Mean :0.2376 Mean :0.3564 Mean :0.5545 Mean :0.604

3rd Qu.:0.0000 3rd Qu.:1.0000 3rd Qu.:1.0000 3rd Qu.:1.000

Max. :1.0000 Max. :1.0000 Max. :1.0000 Max. :1.000

backbone breathes venomous fins

Min. :0.0000 Min. :0.0000 Min. :0.00000 Min. :0.0000

1st Qu.:1.0000 1st Qu.:1.0000 1st Qu.:0.00000 1st Qu.:0.0000

Median :1.0000 Median :1.0000 Median :0.00000 Median :0.0000

Mean :0.8218 Mean :0.7921 Mean :0.07921 Mean :0.1683

3rd Qu.:1.0000 3rd Qu.:1.0000 3rd Qu.:0.00000 3rd Qu.:0.0000

Max. :1.0000 Max. :1.0000 Max. :1.00000 Max. :1.0000

legs tail domestic catsize type

Min. :0.000 Min. :0.0000 Min. :0.0000 Min. :0.0000 Min. :1.000

1st Qu.:2.000 1st Qu.:0.0000 1st Qu.:0.0000 1st Qu.:0.0000 1st Qu.:1.000

Median :4.000 Median :1.0000 Median :0.0000 Median :0.0000 Median :2.000

Mean :2.842 Mean :0.7426 Mean :0.1287 Mean :0.4356 Mean :2.832

3rd Qu.:4.000 3rd Qu.:1.0000 3rd Qu.:0.0000 3rd Qu.:1.0000 3rd Qu.:4.000

Max. :8.000 Max. :1.0000 Max. :1.0000 Max. :1.0000 Max. :7.000

> describe(zoo1)

vars n mean sd median trimmed mad min max range skew kurtosis se

hair 1 101 0.43 0.50 0 0.41 0.00 0 1 1 0.30 -1.93 0.05

feathers 2 101 0.20 0.40 0 0.12 0.00 0 1 1 1.49 0.23 0.04

eggs 3 101 0.58 0.50 1 0.60 0.00 0 1 1 -0.34 -1.91 0.05

milk 4 101 0.41 0.49 0 0.38 0.00 0 1 1 0.38 -1.88 0.05

airborne 5 101 0.24 0.43 0 0.17 0.00 0 1 1 1.21 -0.53 0.04

aquatic 6 101 0.36 0.48 0 0.32 0.00 0 1 1 0.59 -1.67 0.05

predator 7 101 0.55 0.50 1 0.57 0.00 0 1 1 -0.22 -1.97 0.05

toothed 8 101 0.60 0.49 1 0.63 0.00 0 1 1 -0.42 -1.84 0.05

backbone 9 101 0.82 0.38 1 0.90 0.00 0 1 1 -1.66 0.75 0.04

breathes 10 101 0.79 0.41 1 0.86 0.00 0 1 1 -1.42 0.01 0.04

venomous 11 101 0.08 0.27 0 0.00 0.00 0 1 1 3.07 7.50 0.03

fins 12 101 0.17 0.38 0 0.09 0.00 0 1 1 1.75 1.06 0.04

legs 13 101 2.84 2.03 4 2.75 2.97 0 8 8 0.14 -0.67 0.20

tail 14 101 0.74 0.44 1 0.80 0.00 0 1 1 -1.09 -0.81 0.04

domestic 15 101 0.13 0.34 0 0.04 0.00 0 1 1 2.18 2.80 0.03

catsize 16 101 0.44 0.50 0 0.42 0.00 0 1 1 0.26 -1.95 0.05

type 17 101 2.83 2.10 2 2.54 1.48 1 7 6 0.83 -0.77 0.21

> attach(zoo1)

> str(zoo1) # data is in int format

'data.frame': 101 obs. of 17 variables:

$ hair : int 1 1 0 1 1 1 1 0 0 1 ...

$ feathers: int 0 0 0 0 0 0 0 0 0 0 ...

$ eggs : int 0 0 1 0 0 0 0 1 1 0 ...

$ milk : int 1 1 0 1 1 1 1 0 0 1 ...

$ airborne: int 0 0 0 0 0 0 0 0 0 0 ...

$ aquatic : int 0 0 1 0 0 0 0 1 1 0 ...

$ predator: int 1 0 1 1 1 0 0 0 1 0 ...

$ toothed : int 1 1 1 1 1 1 1 1 1 1 ...

$ backbone: int 1 1 1 1 1 1 1 1 1 1 ...

$ breathes: int 1 1 0 1 1 1 1 0 0 1 ...

$ venomous: int 0 0 0 0 0 0 0 0 0 0 ...

$ fins : int 0 0 1 0 0 0 0 1 1 0 ...

$ legs : int 4 4 0 4 4 4 4 0 0 4 ...

$ tail : int 0 1 1 0 1 1 1 1 1 0 ...

$ domestic: int 0 0 0 0 0 0 1 1 0 1 ...

$ catsize : int 1 1 0 1 1 1 1 0 0 0 ...

$ type : int 1 1 4 1 1 1 1 4 4 1 ...

> zoo1$hair <- as.factor(zoo1$hair)

> zoo1$feathers <- as.factor(zoo1$feathers)

> zoo1$eggs <- as.factor(zoo1$eggs)

> zoo1$milk <- as.factor(zoo1$milk)

> zoo1$airborne <- as.factor(zoo1$airborne)

> zoo1$aquatic <- as.factor(zoo1$aquatic)

> zoo1$predator <- as.factor(zoo1$predator)

> zoo1$toothed <- as.factor(zoo1$toothed)

> zoo1$backbone <- as.factor(zoo1$backbone)

> zoo1$breathes <- as.factor(zoo1$breathes)

> zoo1$venomous <- as.factor(zoo1$venomous)

> zoo1$fins <- as.factor(zoo1$fins)

> zoo1$legs <- as.factor(zoo1$legs)

> zoo1$tail <- as.factor(zoo1$tail)

> zoo1$domestic <- as.factor(zoo1$domestic)

> zoo1$catsize <- as.factor(zoo1$catsize)

> zoo1$type <- as.factor(zoo1$type)

> str(zoo1) # data is in fact format

'data.frame': 101 obs. of 17 variables:

$ hair : Factor w/ 2 levels "0","1": 2 2 1 2 2 2 2 1 1 2 ...

$ feathers: Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...

$ eggs : Factor w/ 2 levels "0","1": 1 1 2 1 1 1 1 2 2 1 ...

$ milk : Factor w/ 2 levels "0","1": 2 2 1 2 2 2 2 1 1 2 ...

$ airborne: Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...

$ aquatic : Factor w/ 2 levels "0","1": 1 1 2 1 1 1 1 2 2 1 ...

$ predator: Factor w/ 2 levels "0","1": 2 1 2 2 2 1 1 1 2 1 ...

$ toothed : Factor w/ 2 levels "0","1": 2 2 2 2 2 2 2 2 2 2 ...

$ backbone: Factor w/ 2 levels "0","1": 2 2 2 2 2 2 2 2 2 2 ...

$ breathes: Factor w/ 2 levels "0","1": 2 2 1 2 2 2 2 1 1 2 ...

$ venomous: Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...

$ fins : Factor w/ 2 levels "0","1": 1 1 2 1 1 1 1 2 2 1 ...

$ legs : Factor w/ 6 levels "0","2","4","5",..: 3 3 1 3 3 3 3 1 1 3 ...

$ tail : Factor w/ 2 levels "0","1": 1 2 2 1 2 2 2 2 2 1 ...

$ domestic: Factor w/ 2 levels "0","1": 1 1 1 1 1 1 2 2 1 2 ...

$ catsize : Factor w/ 2 levels "0","1": 2 2 1 2 2 2 2 1 1 1 ...

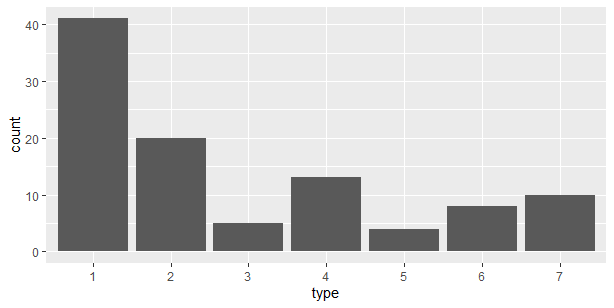
$ type : Factor w/ 7 levels "1","2","3","4",..: 1 1 4 1 1 1 1 4 4 1 ...

> table(zoo1$type)

1 2 3 4 5 6 7

41 20 5 13 4 8 10

> ggplot(zoo1)+geom\_bar(mapping = aes(x=type))



> # splitting of data to train and test

> z <- createDataPartition(type,p=0.8,list = F)

> trn\_z <- zoo1[z,]

> tst\_z <- zoo1[-z,]

> #model building

> set.seed(10)

> model <- knn(trn\_z,tst\_z,k = 7,cl = trn\_z$type)

> summary(model)

1 2 3 4 5 6 7

10 2 0 3 1 3 0

> mean(model==tst\_z$type) # acc =84.2%

[1] 0.6842105

> confusionMatrix(model,tst\_z$type)

Confusion Matrix and Statistics

Reference

Prediction 1 2 3 4 5 6 7

1 10 0 0 0 0 0 0

2 0 2 0 0 0 0 0

3 0 0 0 0 0 0 0

4 0 0 1 1 0 0 1

5 0 0 1 0 0 0 0

6 0 0 0 0 0 0 3

7 0 0 0 0 0 0 0

Overall Statistics

Accuracy : 0.6842

95% CI : (0.4345, 0.8742)

No Information Rate : 0.5263

P-Value [Acc > NIR] : 0.1248

Kappa : 0.5512

Mcnemar's Test P-Value : NA

Statistics by Class:

Class: 1 Class: 2 Class: 3 Class: 4 Class: 5 Class: 6 Class: 7

Sensitivity 1.0000 1.0000 0.0000 1.00000 NA NA 0.0000

Specificity 1.0000 1.0000 1.0000 0.88889 0.94737 0.8421 1.0000

Pos Pred Value 1.0000 1.0000 NaN 0.33333 NA NA NaN

Neg Pred Value 1.0000 1.0000 0.8947 1.00000 NA NA 0.7895

Prevalence 0.5263 0.1053 0.1053 0.05263 0.00000 0.0000 0.2105

Detection Rate 0.5263 0.1053 0.0000 0.05263 0.00000 0.0000 0.0000

Detection Prevalence 0.5263 0.1053 0.0000 0.15789 0.05263 0.1579 0.0000

Balanced Accuracy 1.0000 1.0000 0.5000 0.94444 NA NA 0.5000

> # improving accuracy #boosting

> trn\_acc <- c()

> tst\_acc <- c()

> for(i in seq(1,50,2)){

+ set.seed(100)

+ pred\_knn\_train <- knn(trn\_z,trn\_z,k=i,cl=trn\_z$type)

+ pred\_knn\_test <- knn(trn\_z,tst\_z,k=i,cl=trn\_z$type)

+ trn\_acc <- c(trn\_acc,mean(pred\_knn\_train==trn\_z$type))

+ tst\_acc <- c(tst\_acc,mean(pred\_knn\_test==tst\_z$type))

+ }

> trn\_acc

[1] 1.0000000 0.9390244 0.9024390 0.8902439 0.8902439 0.8414634 0.8048780 0.8048780

[9] 0.8048780 0.7804878 0.7073171 0.7073171 0.6951220 0.6341463 0.6097561 0.5121951

[17] 0.5121951 0.4512195 0.3902439 0.3780488 0.3658537 0.3658537 0.3414634 0.3658537

[25] 0.3780488

> tst\_acc

[1] 1.0000000 0.7368421 0.7368421 0.6842105 0.6842105 0.6842105 0.6842105 0.6842105

[9] 0.6842105 0.6842105 0.6842105 0.6842105 0.6842105 0.6842105 0.6315789 0.6315789

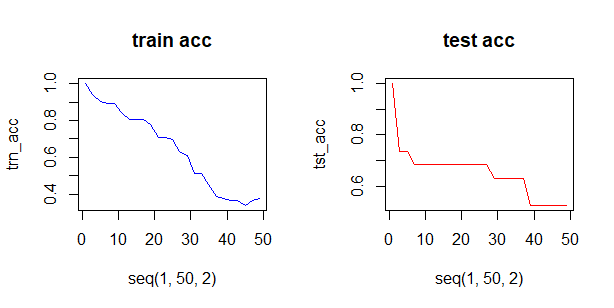
[17] 0.6315789 0.6315789 0.6315789 0.5263158 0.5263158 0.5263158 0.5263158 0.5263158

[25] 0.5263158

> par(mfrow=c(1,2))

> plot(seq(1,50,2),trn\_acc,type = 'l',main = "train acc",col='blue')

> plot(seq(1,50,2),tst\_acc,type = "l",main = "test acc",col='red')



# we have highest test accuracy for k=1

#final model

> final\_zoo <- knn(trn\_z,tst\_z,k=1,cl=trn\_z$type)

> final\_zoo <- knn(trn\_z,tst\_z,k=1,cl=trn\_z$type)

> mean(final\_zoo==tst\_z$type) #acc = 1

[1] 1

> confusionMatrix(final\_zoo,tst\_z$type)

Confusion Matrix and Statistics

Reference

Prediction 1 2 3 4 5 6 7

1 10 0 0 0 0 0 0

2 0 2 0 0 0 0 0

3 0 0 2 0 0 0 0

4 0 0 0 1 0 0 0

5 0 0 0 0 0 0 0

6 0 0 0 0 0 0 0

7 0 0 0 0 0 0 4

Overall Statistics

Accuracy : 1

95% CI : (0.8235, 1)

No Information Rate : 0.5263

P-Value [Acc > NIR] : 5.055e-06

Kappa : 1

Mcnemar's Test P-Value : NA

Statistics by Class:

Class: 1 Class: 2 Class: 3 Class: 4 Class: 5 Class: 6 Class: 7

Sensitivity 1.0000 1.0000 1.0000 1.00000 NA NA 1.0000

Specificity 1.0000 1.0000 1.0000 1.00000 1 1 1.0000

Pos Pred Value 1.0000 1.0000 1.0000 1.00000 NA NA 1.0000

Neg Pred Value 1.0000 1.0000 1.0000 1.00000 NA NA 1.0000

Prevalence 0.5263 0.1053 0.1053 0.05263 0 0 0.2105

Detection Rate 0.5263 0.1053 0.1053 0.05263 0 0 0.2105

Detection Prevalence 0.5263 0.1053 0.1053 0.05263 0 0 0.2105

Balanced Accuracy 1.0000 1.0000 1.0000 1.00000 NA NA 1.0000

> #visualisation of bagging method

> acc\_k <- data.frame(list(train\_acc=trn\_acc,tst\_acc=tst\_acc,k=seq(1,50,2)))

> acc\_k

train\_acc tst\_acc k

1 1.0000000 1.0000000 1

2 0.9390244 0.7368421 3

3 0.9024390 0.7368421 5

4 0.8902439 0.6842105 7

5 0.8902439 0.6842105 9

6 0.8414634 0.6842105 11

7 0.8048780 0.6842105 13

8 0.8048780 0.6842105 15

9 0.8048780 0.6842105 17

10 0.7804878 0.6842105 19

11 0.7073171 0.6842105 21

12 0.7073171 0.6842105 23

13 0.6951220 0.6842105 25

14 0.6341463 0.6842105 27

15 0.6097561 0.6315789 29

16 0.5121951 0.6315789 31

17 0.5121951 0.6315789 33

18 0.4512195 0.6315789 35

19 0.3902439 0.6315789 37

20 0.3780488 0.5263158 39

21 0.3658537 0.5263158 41

22 0.3658537 0.5263158 43

23 0.3414634 0.5263158 45

24 0.3658537 0.5263158 47

25 0.3780488 0.5263158 49

> ggplot(acc\_k,aes(x=k))+

+ geom\_line(aes(y=trn\_acc,color="trn\_acc"))+

+ geom\_line(aes(y=tst\_acc,color="tst\_acc"))

