**Implement a KNN model to classify the animals in to categories**

**Ans:**

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

> head(zoo)

animal.name hair feathers eggs milk airborne aquatic predator toothed backbone

1 aardvark 1 0 0 1 0 0 1 1 1

2 antelope 1 0 0 1 0 0 0 1 1

3 bass 0 0 1 0 0 1 1 1 1

4 bear 1 0 0 1 0 0 1 1 1

5 boar 1 0 0 1 0 0 1 1 1

6 buffalo 1 0 0 1 0 0 0 1 1

breathes venomous fins legs tail domestic catsize type

1 1 0 0 4 0 0 1 1

2 1 0 0 4 1 0 1 1

3 0 0 1 0 1 0 0 4

4 1 0 0 4 0 0 1 1

5 1 0 0 4 1 0 1 1

6 1 0 0 4 1 0 1 1

> table(zoo$type)

1 2 3 4 5 6 7

41 20 5 13 4 8 10

> str(zoo)

'data.frame': 101 obs. of 18 variables:

$ animal.name: Factor w/ 100 levels "aardvark","antelope",..: 1 2 3 4 5 6 7 8 9 10 ...

$ 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 ...

**Structure of variable ‘type’ is int, need to convert it to factor**

> zoo$type <- factor(zoo$type)

> str(zoo)

'data.frame': 101 obs. of 18 variables:

$ animal.name: Factor w/ 100 levels "aardvark","antelope",..: 1 2 3 4 5 6 7 8 9 10 ...

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

**Table or proportation of enteries in the datasets to check Biasness.**

> round(prop.table(table(zoo$type))\*100,1)

1 2 3 4 5 6 7

40.6 19.8 5.0 12.9 4.0 7.9 9.9

**‘Type’ Variable is not highly biased towards specific glass type as observed above.**

> summary(zoo)

animal.name hair feathers eggs milk

frog : 2 Min. :0.0000 Min. :0.000 Min. :0.0000 Min. :0.0000

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

antelope: 1 Median :0.0000 Median :0.000 Median :1.0000 Median :0.0000

bass : 1 Mean :0.4257 Mean :0.198 Mean :0.5842 Mean :0.4059

bear : 1 3rd Qu.:1.0000 3rd Qu.:0.000 3rd Qu.:1.0000 3rd Qu.:1.0000

boar : 1 Max. :1.0000 Max. :1.000 Max. :1.0000 Max. :1.0000

(Other) :94

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 1:41

1st Qu.:2.000 1st Qu.:0.0000 1st Qu.:0.0000 1st Qu.:0.0000 2:20

Median :4.000 Median :1.0000 Median :0.0000 Median :0.0000 3: 5

Mean :2.842 Mean :0.7426 Mean :0.1287 Mean :0.4356 4:13

3rd Qu.:4.000 3rd Qu.:1.0000 3rd Qu.:0.0000 3rd Qu.:1.0000 5: 4

Max. :8.000 Max. :1.0000 Max. :1.0000 Max. :1.0000 6: 8

7:10

**Creating a function to normalize the data**

>norm <- function(x){

return((x-min(x))/(max(x)-min(x)))

}

**Applying normalization function to the dataset.**

**Creating 70:30 train and test datasets**

> zoo\_train <- zoo\_norm[1:70,]

> zoo\_test <- zoo\_norm[71:101,]

**Get labels for training and test datasets**

> zoo\_train\_labels <- zoo[1:70,1]

> zoo\_test\_labels <- zoo[71:101,1]

**Building the KNN model on training dataset and then test on test dataset**

>test\_acc <- NULL

>train\_acc <- NULL

>for (i in seq(3,70,2))

{

train\_zoo\_pred <- knn(train=zoo\_train,test=zoo\_train,cl=zoo\_train\_labels,k=i)

train\_acc <- c(train\_acc,mean(train\_zoo\_pred==zoo\_train\_labels))

test\_zoo\_pred <- knn(train = zoo\_train, test = zoo\_test, cl = zoo\_train\_labels, k=i)

test\_acc <- c(test\_acc,mean(test\_zoo\_pred==zoo\_test\_labels))

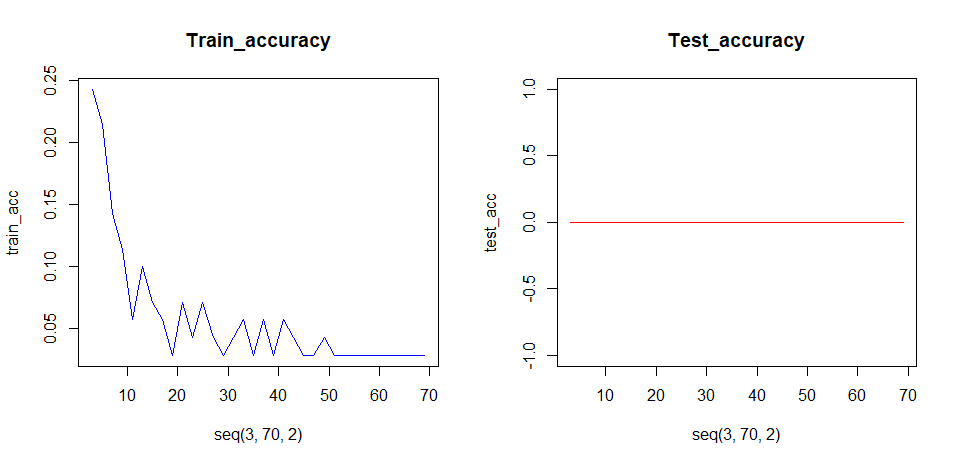
}

**Testing Accuracy**

>par(mfrow=c(1,2)) # c(1,2) => indicates 1 row and 2 columns

>plot(seq(3,70,2),train\_acc,type="l",main="Train\_accuracy",col="blue")

>plot(seq(3,70,2),test\_acc,type="l",main="Test\_accuracy",col="red")



**For K= 3, Accuracy is good**

> zoo\_pred <- knn(train = zoo\_train, test = zoo\_test, cl = zoo\_train\_labels, k=3)

**So we will Deploy the model with k=3 on train and test data.**