K-Nearest Neighbour

K-NN

Idea Behind K-NN

Birds of the Same Feather Flock Together

Courtesy: www.understandingsociety.ac.uk/2013/07/26/do-birds-of-a-feather-flock-together

Courtesy: http://positivity360.com/post-2/

K - Nearest Neighbors

- In k-nearest neighbors method, the classifier identifies k observations in the training dataset that are similar to a new record that we wish to classify.
- The classifier looks for records in our training data that are similar or "near" the record to be classified in the predictor space (i.e., records that have values close to X1, X2, . . . , Xp).
- Then, based on the classes to which those proximate records belong, we assign a class to the record that we want to classify.

Distance Method

- For record i we have the vector of p measurements (xi1, xi2, . . . , xip), while for record j we have the vector of measurements (xj1, xj2, . . . , xjp).
- The most popular distance measure is the Euclidean distance, dij , which between two cases, i and j, is defined by

$$d_{ij} = \sqrt{(x_{i1}-x_{j1})^2+(x_{i2}-x_{j2})^2+\ldots+(x_{ip}-x_{jp})^2}$$

Other Distance Measures

- Numerical Data
  - Correlation-based similarity
  - Statistical distance (also called Mahalanobis distance)
  - Manhattan distance ("city block")
  - Maximum coordinate distance
- Categorical Data
  - Matching coefficient: (a + d)/p
  - Jaquard's coefficient: d/(b+c+d)

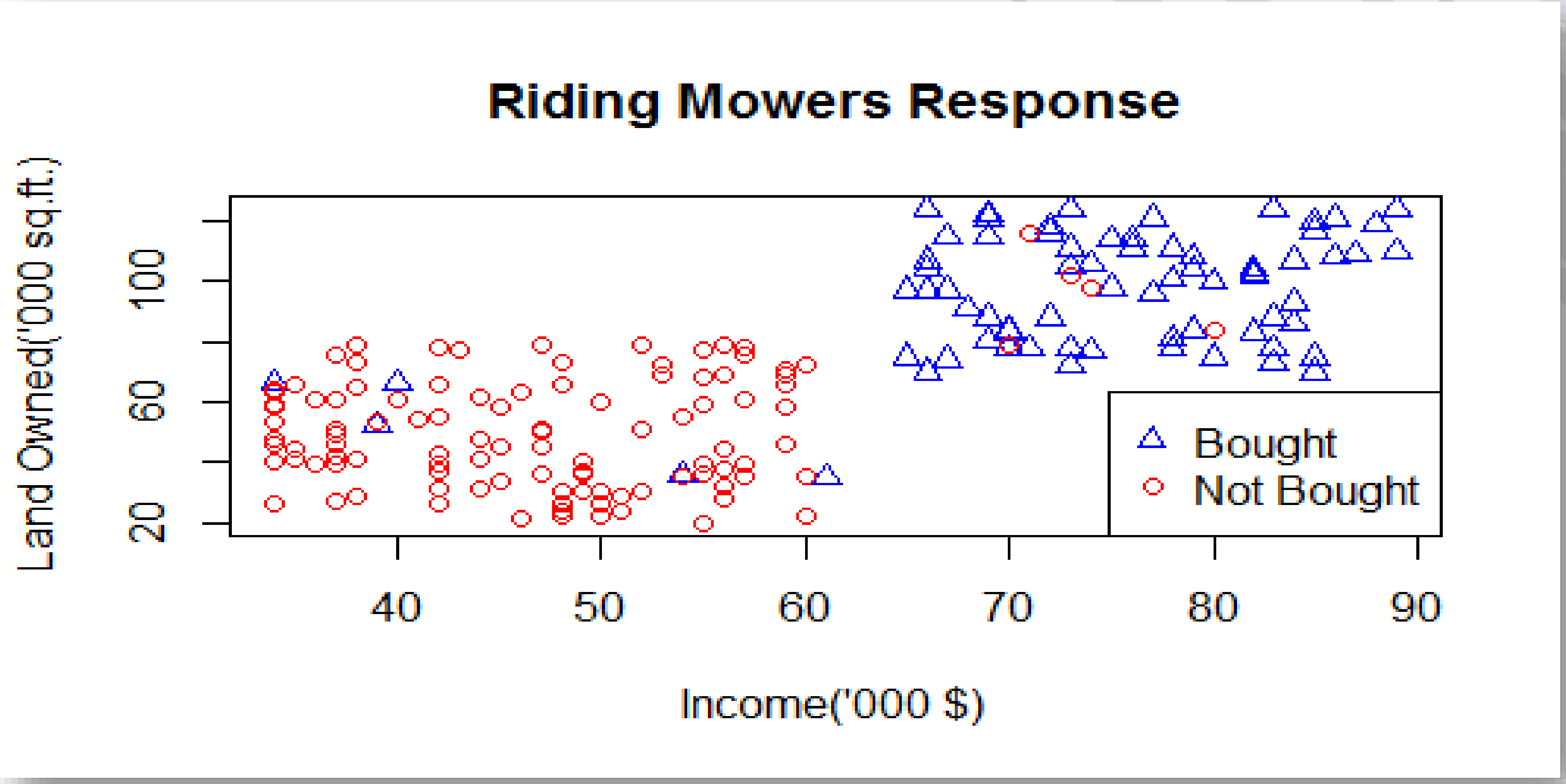
K - NN

- The k-nearest neighbors algorithm is a classification method that does not make assumptions about the form of the relationship between the response (Y) and the predictors X1,X2, . . .,Xp.
- This is a nonparametric method because it does not involve estimation of parameters as against the methods like linear regression.

Example: Riding Mowers

- A riding-mower manufacturer **MOW-EASE** took part in a Industrial Exhibition in which it got an opportunity to show a demo of its product to 180 different audience.
- The land owned by each of the audience and their approximate income have been recorded in the file RidingMowers.csv

Visualizing the Data



- Here we see that the response has some pattern of farness or nearness

Nearest Observations: K=1

- Consider a person with Income as $ 70,000 and Lot size as 100,000 sq. ft.
- By Euclidean Distance Method, the nearest one observation is the 136th observation.

136	73	102	Not Bought

- As we can see here, that 136th observation person has not bought in spite of showing him the product demo. Hence we can conclude that the person with Income as $ 70,000 and Lot size as 100,000 sq. ft. won’t buy.

Nearest Observations: K=3

- By Euclidean Distance Method, the nearest three observations are 136^{th}, 116^{th} and 141^{st}.

136	73	102	Not Bought
116	67	97	Bought
141	74	98	Not Bought

- As we can see here, that 2 have not bought and 1 has bought in spite of showing him the product demo. Hence we can conclude that the person with Income as $ 70,000 and Lot size as 100,000 sq. ft. won't buy.

Nearest Observations: K=5

- By Euclidean Distance Method, the nearest three observations are 136th, 137th, 116th, 143rd and 141st.

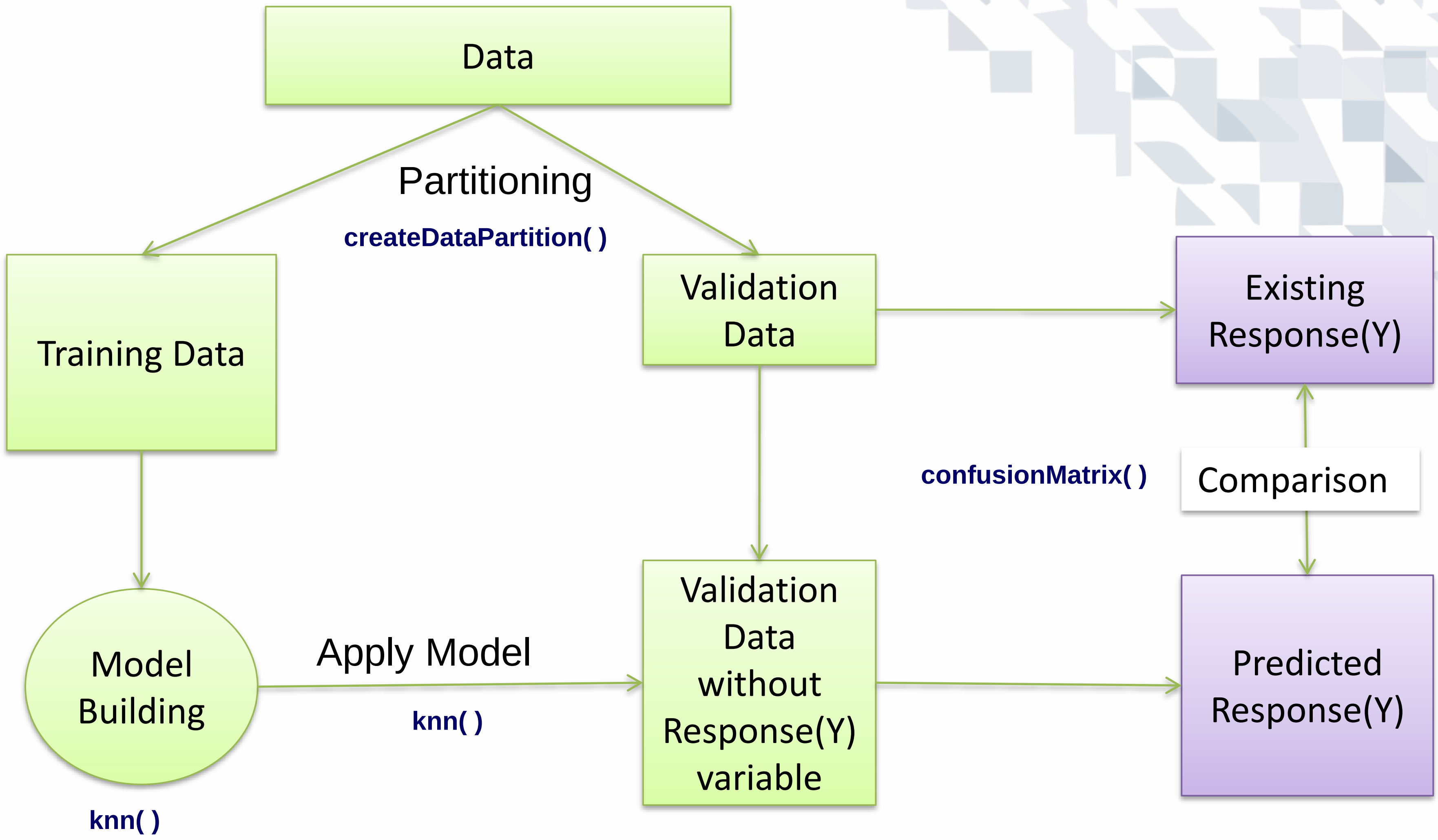
116	67	97	Bought
136	73	102	Not Bought
137	73	105	Bought
141	74	98	Not Bought
143	75	98	Bought

- As we can see here, that 2 have not bought and 3 have bought in spite of showing him the product demo. Hence we can conclude that the person with Income as $ 70,000 and Lot size as 100,000 sq. ft. will buy.

K-NN in R

- K-NN can be implemented in different ways in R. We will cover the two ways:
  - By package class
  - By package caret

K-NN Classifier with package class



knn() in package class

Syntax:

knn(training, validation, cl, k, …)

Where

training : matrix or data frame of predictors in training set

validation : matrix or data frame of predictors in validation set

cl : factor vector of response variable in training set

k : number of neighbors considered

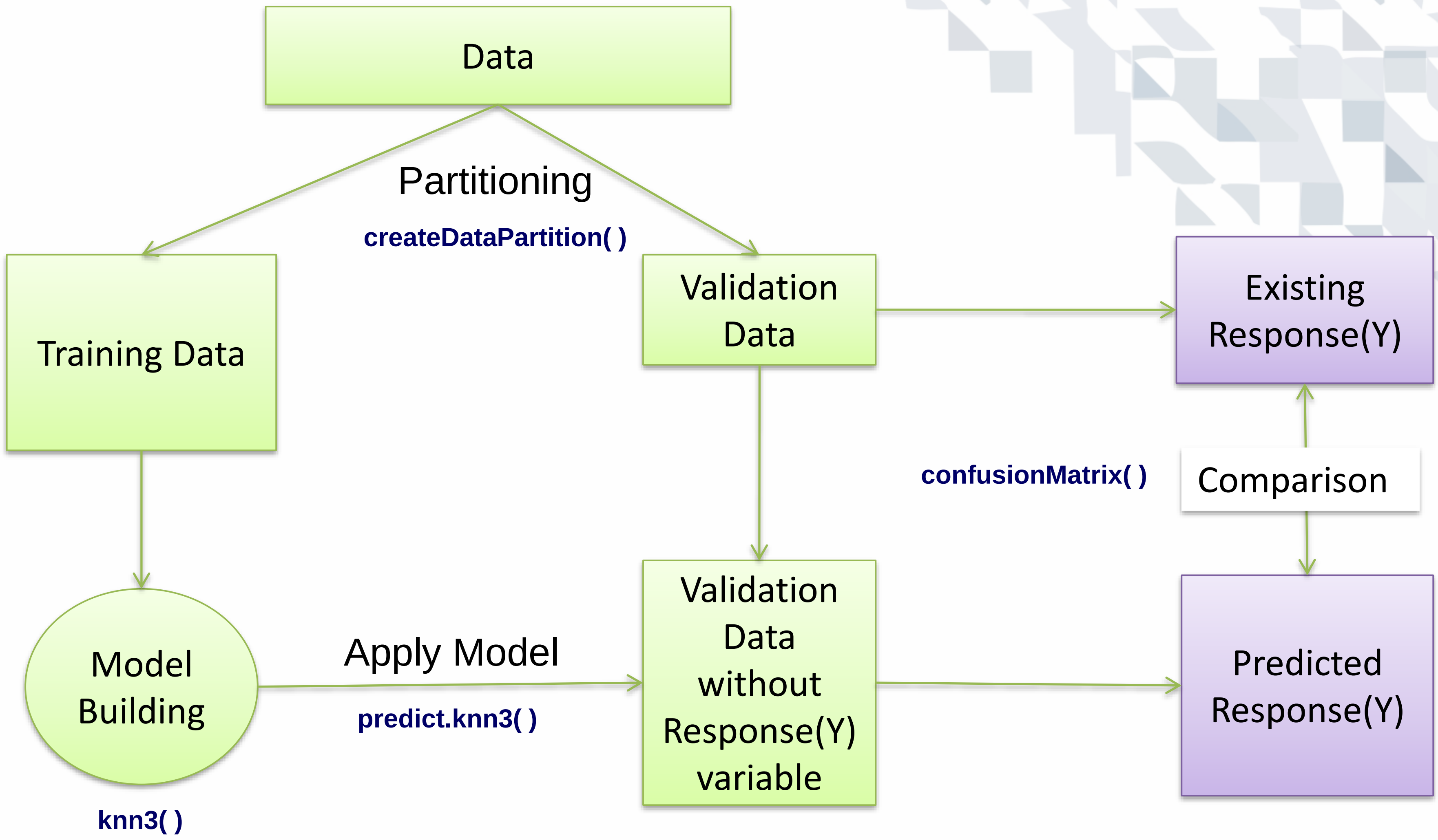
Program and Output

```
library(caret)
set.seed(1992)
intrain<-createDataPartition(y=mowers$Response,p=0.7,list=FALSE)


trainingWOY <- mowers[intrain,-3]
validationWOY <- mowers[-intrain,-3]

YofTraining <- mowers[intrain,3]
YofValidation <- mowers[-intrain,3]
```

```
library(class)

knn1.pred=knn(trainingWOY,validationWOY,YofTraining,k=1)
tbl_1 <- table(knn1.pred , YofValidation)
confusionMatrix( tbl_1 )
```

```
Confusion Matrix and Statistics

            YofValidation
knn1.pred     Bought Not Bought
  Bought          16          2
  Not Bought       5         30

               Accuracy : 0.8679
                 95% CI : (0.7466, 0.9452)
    No Information Rate : 0.6038
    P-Value [Acc > NIR] : 2.513e-05

                  Kappa : 0.717
 Mcnemar's Test P-Value : 0.4497

            Sensitivity : 0.7619
            Specificity : 0.9375
         Pos Pred Value : 0.8889
         Neg Pred Value : 0.8571
             Prevalence : 0.3962
         Detection Rate : 0.3019
   Detection Prevalence : 0.3396
      Balanced Accuracy : 0.8497

       'Positive' Class : Bought
```

K-NN Classifier with package caret



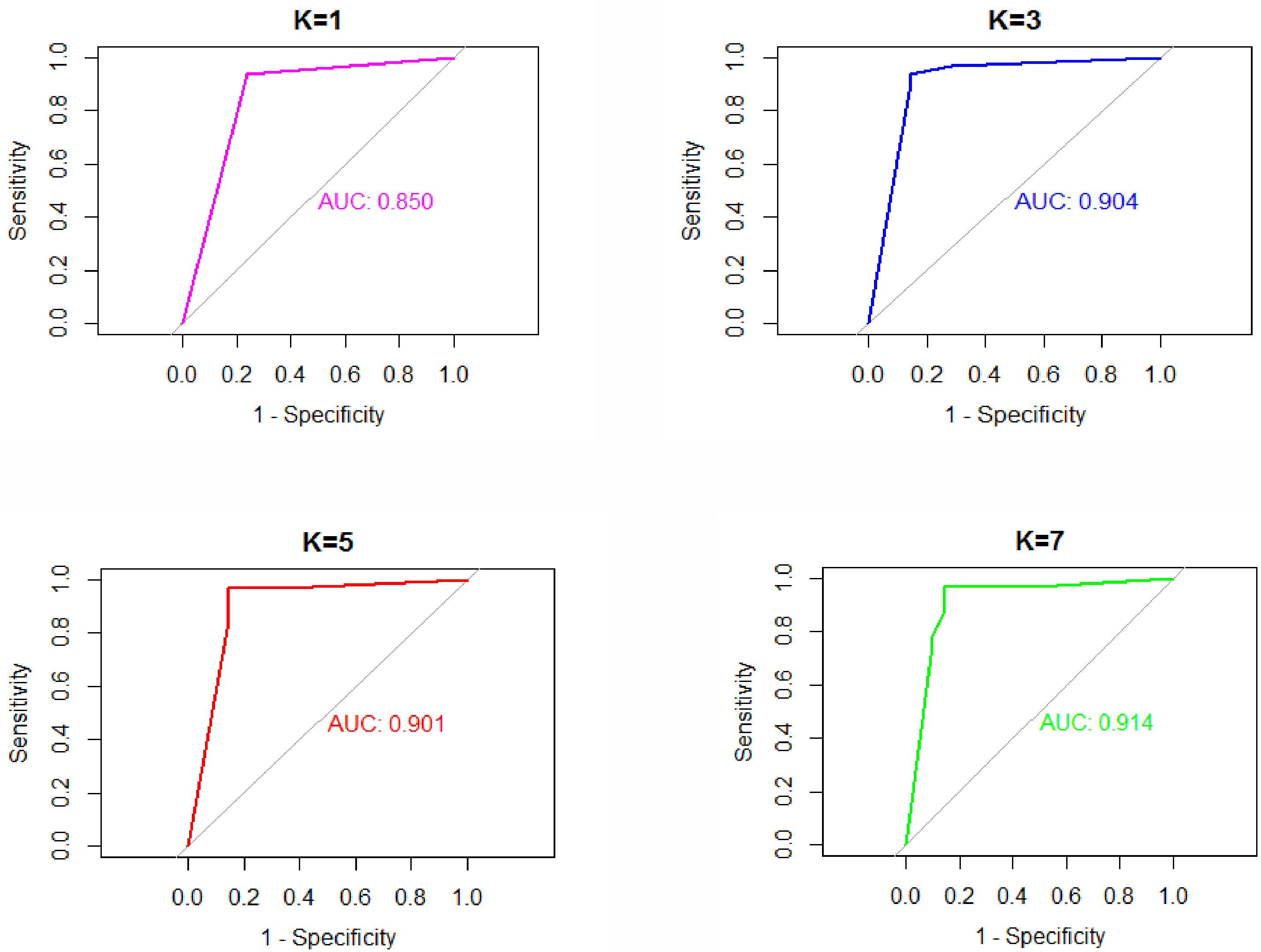
Sane's STATS Academy of Statistics

Program and Output

```r
set.seed(1992)
intrain<-createDataPartition(y=mowers$Response,p=0.7,list=FALSE)

training <- mowers[intrain, ]
validation <- mowers[-intrain, ]


# Using knn3 function

fitKNN1 <- knn3(Response ~ .,data=training, k=1)
pred.knn1 <- predict.knn3(fitKNN1,newdata=validation,type = "class")
tbl_1 <- table(pred.knn1 , validation$Response  )
confusionMatrix( tbl_1 )
```

```
Confusion Matrix and Statistics


pred.knn1     Bought Not Bought
  Bought          16          2
  Not Bought       5         30

               Accuracy : 0.8679
                 95% CI : (0.7466, 0.9452)
    No Information Rate : 0.6038
    P-Value [Acc > NIR] : 2.513e-05

                  Kappa : 0.717
 Mcnemar's Test P-Value : 0.4497

            Sensitivity : 0.7619
            Specificity : 0.9375
         Pos Pred Value : 0.8889
         Neg Pred Value : 0.8571
             Prevalence : 0.3962
         Detection Rate : 0.3019
   Detection Prevalence : 0.3396
      Balanced Accuracy : 0.8497

       'Positive' Class : Bought
```

ROC Curves


K=1
AUC: 0.850
K=3
AUC: 0.904
K=5
AUC: 0.901
K=7
AUC: 0.914
Sensitivity
1 - Specificity
0.0 0.2 0.4 0.6 0.8 1.0

All in One ROC