Chapter 4 Lab

Noah Foilb

4/3/2022

#Libraries

library(ISLR)

## Warning: package 'ISLR' was built under R version 4.0.4

library(caret)

## Warning: package 'caret' was built under R version 4.0.4

## Loading required package: lattice

## Loading required package: ggplot2

## Warning: package 'ggplot2' was built under R version 4.0.5

library(dplyr)

## Warning: package 'dplyr' was built under R version 4.0.5

##   
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':  
##   
## filter, lag

## The following objects are masked from 'package:base':  
##   
## intersect, setdiff, setequal, union

library(MASS)

##   
## Attaching package: 'MASS'

## The following object is masked from 'package:dplyr':  
##   
## select

# 10

# Data

nrow(Weekly)

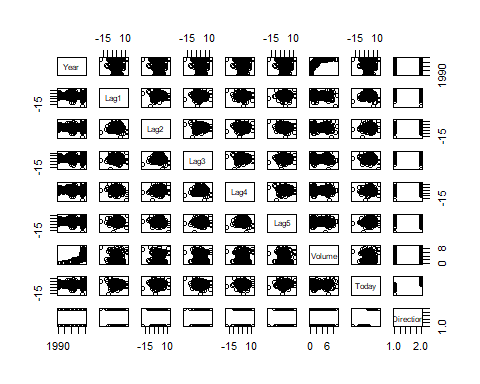
## [1] 1089

summary(Weekly)

## Year Lag1 Lag2 Lag3   
## Min. :1990 Min. :-18.1950 Min. :-18.1950 Min. :-18.1950   
## 1st Qu.:1995 1st Qu.: -1.1540 1st Qu.: -1.1540 1st Qu.: -1.1580   
## Median :2000 Median : 0.2410 Median : 0.2410 Median : 0.2410   
## Mean :2000 Mean : 0.1506 Mean : 0.1511 Mean : 0.1472   
## 3rd Qu.:2005 3rd Qu.: 1.4050 3rd Qu.: 1.4090 3rd Qu.: 1.4090   
## Max. :2010 Max. : 12.0260 Max. : 12.0260 Max. : 12.0260   
## Lag4 Lag5 Volume Today   
## Min. :-18.1950 Min. :-18.1950 Min. :0.08747 Min. :-18.1950   
## 1st Qu.: -1.1580 1st Qu.: -1.1660 1st Qu.:0.33202 1st Qu.: -1.1540   
## Median : 0.2380 Median : 0.2340 Median :1.00268 Median : 0.2410   
## Mean : 0.1458 Mean : 0.1399 Mean :1.57462 Mean : 0.1499   
## 3rd Qu.: 1.4090 3rd Qu.: 1.4050 3rd Qu.:2.05373 3rd Qu.: 1.4050   
## Max. : 12.0260 Max. : 12.0260 Max. :9.32821 Max. : 12.0260   
## Direction   
## Down:484   
## Up :605   
##   
##   
##   
##

# A)

pairs(Weekly)



# A good amount of the data seems to be highly correlated. While this is great, we should keep this in mind as to not overfit the model with confounding factors

# B)

M1 <- glm(Direction ~ Lag1 + Lag2 + Lag3 + Lag4 + Lag5 + Volume, data = Weekly, family = "binomial")  
summary(M1)

##   
## Call:  
## glm(formula = Direction ~ Lag1 + Lag2 + Lag3 + Lag4 + Lag5 +   
## Volume, family = "binomial", data = Weekly)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.6949 -1.2565 0.9913 1.0849 1.4579   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 0.26686 0.08593 3.106 0.0019 \*\*  
## Lag1 -0.04127 0.02641 -1.563 0.1181   
## Lag2 0.05844 0.02686 2.175 0.0296 \*   
## Lag3 -0.01606 0.02666 -0.602 0.5469   
## Lag4 -0.02779 0.02646 -1.050 0.2937   
## Lag5 -0.01447 0.02638 -0.549 0.5833   
## Volume -0.02274 0.03690 -0.616 0.5377   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 1496.2 on 1088 degrees of freedom  
## Residual deviance: 1486.4 on 1082 degrees of freedom  
## AIC: 1500.4  
##   
## Number of Fisher Scoring iterations: 4

confint(M1)

## Waiting for profiling to be done...

## 2.5 % 97.5 %  
## (Intercept) 0.098808746 0.43580101  
## Lag1 -0.093477110 0.01029269  
## Lag2 0.006197597 0.11169774  
## Lag3 -0.068653910 0.03604309  
## Lag4 -0.079952378 0.02401603  
## Lag5 -0.066495108 0.03711989  
## Volume -0.095051949 0.04979338

# they are all insignificant as their \_ values are greater than .05 except Lag2, and their 95% confit intervals pass over 0 except Lag2. I would say this is because they are all highly correlated with eachother.

# C)

Prob <- predict(M1, Weekly, type = "response")  
Pred <- rep("Down", 1089)  
Pred[Prob>.5] = "Up"  
Pred <- as.factor(Pred)  
confusionMatrix(Pred,Weekly$Direction)

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction Down Up  
## Down 54 48  
## Up 430 557  
##   
## Accuracy : 0.5611   
## 95% CI : (0.531, 0.5908)  
## No Information Rate : 0.5556   
## P-Value [Acc > NIR] : 0.369   
##   
## Kappa : 0.035   
##   
## Mcnemar's Test P-Value : <2e-16   
##   
## Sensitivity : 0.11157   
## Specificity : 0.92066   
## Pos Pred Value : 0.52941   
## Neg Pred Value : 0.56434   
## Prevalence : 0.44444   
## Detection Rate : 0.04959   
## Detection Prevalence : 0.09366   
## Balanced Accuracy : 0.51612   
##   
## 'Positive' Class : Down   
##

# We can see that this Model has an extremely high Specitivity and not the best accuracy. What we can also tell is that it tends to Guess Up more than Down and is pretty 50/50 for each guess. That is to be expected as the model used for this confusion matrix wasnt statistically significant.

# D)

Train <- Weekly %>%  
 filter(Year != 2009 & Year!= 2010)  
   
Test <- Weekly %>%  
 filter(Year == 2009 | Year == 2010)

M2 <- glm(Direction ~ Lag2, data = Train, family = "binomial")  
summary(M2)

##   
## Call:  
## glm(formula = Direction ~ Lag2, family = "binomial", data = Train)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.536 -1.264 1.021 1.091 1.368   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 0.20326 0.06428 3.162 0.00157 \*\*  
## Lag2 0.05810 0.02870 2.024 0.04298 \*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 1354.7 on 984 degrees of freedom  
## Residual deviance: 1350.5 on 983 degrees of freedom  
## AIC: 1354.5  
##   
## Number of Fisher Scoring iterations: 4

confint(M2)

## Waiting for profiling to be done...

## 2.5 % 97.5 %  
## (Intercept) 0.077479287 0.3295391  
## Lag2 0.002300791 0.1150942

Prob <- predict(M2, Test, type = "response")  
Pred <- rep("Down", 104)  
Pred[Prob>.5] = "Up"  
Pred <- as.factor(Pred)  
confusionMatrix(Pred,Test$Direction)

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction Down Up  
## Down 9 5  
## Up 34 56  
##   
## Accuracy : 0.625   
## 95% CI : (0.5247, 0.718)  
## No Information Rate : 0.5865   
## P-Value [Acc > NIR] : 0.2439   
##   
## Kappa : 0.1414   
##   
## Mcnemar's Test P-Value : 7.34e-06   
##   
## Sensitivity : 0.20930   
## Specificity : 0.91803   
## Pos Pred Value : 0.64286   
## Neg Pred Value : 0.62222   
## Prevalence : 0.41346   
## Detection Rate : 0.08654   
## Detection Prevalence : 0.13462   
## Balanced Accuracy : 0.56367   
##   
## 'Positive' Class : Down   
##

# This model is more accurate than the other model as it is more statistically significant

# E)

M3 <- lda(Direction ~ Lag2, data = Train)  
summary(M3)

## Length Class Mode   
## prior 2 -none- numeric   
## counts 2 -none- numeric   
## means 2 -none- numeric   
## scaling 1 -none- numeric   
## lev 2 -none- character  
## svd 1 -none- numeric   
## N 1 -none- numeric   
## call 3 -none- call   
## terms 3 terms call   
## xlevels 0 -none- list

Prob <- predict(M3, Test, type = "response")  
Pred <- rep("Down", 104)  
Pred[Prob$posterior[,1]>.4] = "Up"  
Pred <- as.factor(Pred)  
confusionMatrix(Pred,Test$Direction)

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction Down Up  
## Down 3 10  
## Up 40 51  
##   
## Accuracy : 0.5192   
## 95% CI : (0.4191, 0.6183)  
## No Information Rate : 0.5865   
## P-Value [Acc > NIR] : 0.9316   
##   
## Kappa : -0.105   
##   
## Mcnemar's Test P-Value : 4.11e-05   
##   
## Sensitivity : 0.06977   
## Specificity : 0.83607   
## Pos Pred Value : 0.23077   
## Neg Pred Value : 0.56044   
## Prevalence : 0.41346   
## Detection Rate : 0.02885   
## Detection Prevalence : 0.12500   
## Balanced Accuracy : 0.45292   
##   
## 'Positive' Class : Down   
##

# With an accuracy of 52.88% it is not better nor more significant than the other model.

# F)

M3 <- qda(Direction ~ Lag2, data = Train)  
summary(M3)

## Length Class Mode   
## prior 2 -none- numeric   
## counts 2 -none- numeric   
## means 2 -none- numeric   
## scaling 2 -none- numeric   
## ldet 2 -none- numeric   
## lev 2 -none- character  
## N 1 -none- numeric   
## call 3 -none- call   
## terms 3 terms call   
## xlevels 0 -none- list

Prob <- predict(M3, Test, type = "response")  
Pred <- rep("Down", 104)  
Pred[as.numeric(Prob$posterior[,1])>.44] = "Up"  
Pred <- as.factor(Pred)  
confusionMatrix(Pred,Test$Direction)

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction Down Up  
## Down 19 24  
## Up 24 37  
##   
## Accuracy : 0.5385   
## 95% CI : (0.438, 0.6367)  
## No Information Rate : 0.5865   
## P-Value [Acc > NIR] : 0.8631   
##   
## Kappa : 0.0484   
##   
## Mcnemar's Test P-Value : 1.0000   
##   
## Sensitivity : 0.4419   
## Specificity : 0.6066   
## Pos Pred Value : 0.4419   
## Neg Pred Value : 0.6066   
## Prevalence : 0.4135   
## Detection Rate : 0.1827   
## Detection Prevalence : 0.4135   
## Balanced Accuracy : 0.5242   
##   
## 'Positive' Class : Down   
##

# This model isnt statisitically significant as well, this model is the worst model so far

# G)

Knn <- knn3(Direction ~ Lag2, data = Train, k = 1)  
summary(Knn)

## Length Class Mode   
## learn 2 -none- list   
## k 1 -none- numeric  
## terms 3 terms call   
## xlevels 0 -none- list   
## theDots 0 -none- list

Pred <- predict(Knn, Test, type = "class")  
confusionMatrix(Pred,Test$Direction)

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction Down Up  
## Down 21 30  
## Up 22 31  
##   
## Accuracy : 0.5   
## 95% CI : (0.4003, 0.5997)  
## No Information Rate : 0.5865   
## P-Value [Acc > NIR] : 0.9700   
##   
## Kappa : -0.0033   
##   
## Mcnemar's Test P-Value : 0.3317   
##   
## Sensitivity : 0.4884   
## Specificity : 0.5082   
## Pos Pred Value : 0.4118   
## Neg Pred Value : 0.5849   
## Prevalence : 0.4135   
## Detection Rate : 0.2019   
## Detection Prevalence : 0.4904   
## Balanced Accuracy : 0.4983   
##   
## 'Positive' Class : Down   
##

# With a .5 Accuracy, this model isnt better than the other models as well.

# H)

# The GLM had by far the best accuracy by having an accuracy of over 10% of any other model. NOt only this but the missclassification rate is the lowest.

# G)

# Different Knn values  
Knn <- knn3(Direction ~ Lag2, data = Train, k = 2)  
summary(Knn)

## Length Class Mode   
## learn 2 -none- list   
## k 1 -none- numeric  
## terms 3 terms call   
## xlevels 0 -none- list   
## theDots 0 -none- list

Pred <- predict(Knn, Test, type = "class")  
confusionMatrix(Pred,Test$Direction)

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction Down Up  
## Down 23 29  
## Up 20 32  
##   
## Accuracy : 0.5288   
## 95% CI : (0.4285, 0.6275)  
## No Information Rate : 0.5865   
## P-Value [Acc > NIR] : 0.9017   
##   
## Kappa : 0.0577   
##   
## Mcnemar's Test P-Value : 0.2531   
##   
## Sensitivity : 0.5349   
## Specificity : 0.5246   
## Pos Pred Value : 0.4423   
## Neg Pred Value : 0.6154   
## Prevalence : 0.4135   
## Detection Rate : 0.2212   
## Detection Prevalence : 0.5000   
## Balanced Accuracy : 0.5297   
##   
## 'Positive' Class : Down   
##

Knn <- knn3(Direction ~ Lag2, data = Train, k = 3)  
summary(Knn)

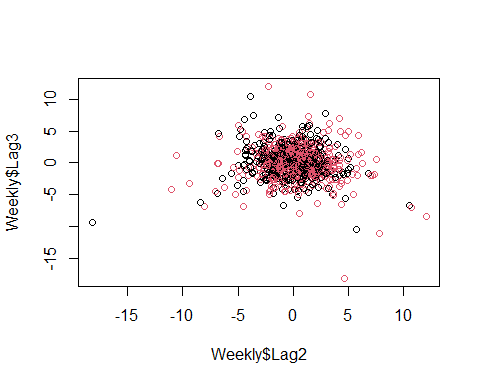
## Length Class Mode   
## learn 2 -none- list   
## k 1 -none- numeric  
## terms 3 terms call   
## xlevels 0 -none- list   
## theDots 0 -none- list

Pred <- predict(Knn, Test, type = "class")  
confusionMatrix(Pred,Test$Direction)

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction Down Up  
## Down 16 19  
## Up 27 42  
##   
## Accuracy : 0.5577   
## 95% CI : (0.457, 0.655)  
## No Information Rate : 0.5865   
## P-Value [Acc > NIR] : 0.7579   
##   
## Kappa : 0.0623   
##   
## Mcnemar's Test P-Value : 0.3020   
##   
## Sensitivity : 0.3721   
## Specificity : 0.6885   
## Pos Pred Value : 0.4571   
## Neg Pred Value : 0.6087   
## Prevalence : 0.4135   
## Detection Rate : 0.1538   
## Detection Prevalence : 0.3365   
## Balanced Accuracy : 0.5303   
##   
## 'Positive' Class : Down   
##

# Different Knn values fo not change much.

plot(Weekly$Lag2,Weekly$Lag3, col = Weekly$Direction)



# Doesnt seem to have a correlation with Lag.

# 13

# Dataset

summary(Boston)

## crim zn indus chas   
## Min. : 0.00632 Min. : 0.00 Min. : 0.46 Min. :0.00000   
## 1st Qu.: 0.08205 1st Qu.: 0.00 1st Qu.: 5.19 1st Qu.:0.00000   
## Median : 0.25651 Median : 0.00 Median : 9.69 Median :0.00000   
## Mean : 3.61352 Mean : 11.36 Mean :11.14 Mean :0.06917   
## 3rd Qu.: 3.67708 3rd Qu.: 12.50 3rd Qu.:18.10 3rd Qu.:0.00000   
## Max. :88.97620 Max. :100.00 Max. :27.74 Max. :1.00000   
## nox rm age dis   
## Min. :0.3850 Min. :3.561 Min. : 2.90 Min. : 1.130   
## 1st Qu.:0.4490 1st Qu.:5.886 1st Qu.: 45.02 1st Qu.: 2.100   
## Median :0.5380 Median :6.208 Median : 77.50 Median : 3.207   
## Mean :0.5547 Mean :6.285 Mean : 68.57 Mean : 3.795   
## 3rd Qu.:0.6240 3rd Qu.:6.623 3rd Qu.: 94.08 3rd Qu.: 5.188   
## Max. :0.8710 Max. :8.780 Max. :100.00 Max. :12.127   
## rad tax ptratio black   
## Min. : 1.000 Min. :187.0 Min. :12.60 Min. : 0.32   
## 1st Qu.: 4.000 1st Qu.:279.0 1st Qu.:17.40 1st Qu.:375.38   
## Median : 5.000 Median :330.0 Median :19.05 Median :391.44   
## Mean : 9.549 Mean :408.2 Mean :18.46 Mean :356.67   
## 3rd Qu.:24.000 3rd Qu.:666.0 3rd Qu.:20.20 3rd Qu.:396.23   
## Max. :24.000 Max. :711.0 Max. :22.00 Max. :396.90   
## lstat medv   
## Min. : 1.73 Min. : 5.00   
## 1st Qu.: 6.95 1st Qu.:17.02   
## Median :11.36 Median :21.20   
## Mean :12.65 Mean :22.53   
## 3rd Qu.:16.95 3rd Qu.:25.00   
## Max. :37.97 Max. :50.00

Boston <- Boston  
?Boston

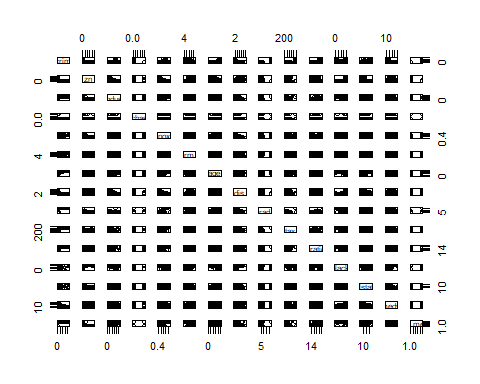
## starting httpd help server ... done

# Split the dataset by the median of crime

median\_crime = median(Boston$crim)  
crim\_median <- rep(0, 506)  
crim\_median[Boston$crim > median\_crime] = 1  
crim\_median <- as.factor(crim\_median)  
Boston <- data.frame(Boston, crim\_median)

# Look into the data

pairs(Boston)



# Train Test split

set.seed(100)  
train <- Boston %>% sample\_frac(size = 0.75)  
test <- Boston %>% setdiff(train)

# Glm

M1 <- glm(crim\_median~nox + medv + dis, data = train, family = binomial)  
summary(M1)

##   
## Call:  
## glm(formula = crim\_median ~ nox + medv + dis, family = binomial,   
## data = train)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -2.0450 -0.4449 0.0044 0.3039 2.3625   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -23.00045 3.54291 -6.492 8.47e-11 \*\*\*  
## nox 37.81787 5.19711 7.277 3.42e-13 \*\*\*  
## medv 0.06248 0.02722 2.296 0.0217 \*   
## dis 0.31773 0.14784 2.149 0.0316 \*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 526.7 on 379 degrees of freedom  
## Residual deviance: 244.4 on 376 degrees of freedom  
## AIC: 252.4  
##   
## Number of Fisher Scoring iterations: 6

# Get rid of Dis  
M1 <- glm(crim\_median~nox + medv, data = train, family = binomial)  
summary(M1)

##   
## Call:  
## glm(formula = crim\_median ~ nox + medv, family = binomial, data = train)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -2.07329 -0.44376 0.01091 0.31524 2.57480   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -17.58588 2.14915 -8.183 2.77e-16 \*\*\*  
## nox 30.35283 3.34259 9.081 < 2e-16 \*\*\*  
## medv 0.05045 0.02571 1.963 0.0497 \*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 526.70 on 379 degrees of freedom  
## Residual deviance: 248.84 on 377 degrees of freedom  
## AIC: 254.84  
##   
## Number of Fisher Scoring iterations: 6

# COnfusion Matrix

Prob <- predict(M1, test, type = "response")  
Pred <- rep(0, 126)  
Pred[Prob>.35] = 1  
Pred <- as.factor(Pred)  
confusionMatrix(Pred,test$crim\_median)

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction 0 1  
## 0 60 4  
## 1 6 56  
##   
## Accuracy : 0.9206   
## 95% CI : (0.8589, 0.9613)  
## No Information Rate : 0.5238   
## P-Value [Acc > NIR] : <2e-16   
##   
## Kappa : 0.8411   
##   
## Mcnemar's Test P-Value : 0.7518   
##   
## Sensitivity : 0.9091   
## Specificity : 0.9333   
## Pos Pred Value : 0.9375   
## Neg Pred Value : 0.9032   
## Prevalence : 0.5238   
## Detection Rate : 0.4762   
## Detection Prevalence : 0.5079   
## Balanced Accuracy : 0.9212   
##   
## 'Positive' Class : 0   
##

# THis is an insane model. Just looking at 92.06% is an insane

# lda

lda <- lda(crim\_median~nox + medv, data = train)  
Prob <- predict(lda, test, type = "Response")  
Pred <- rep(0, 126)  
Pred[Prob$posterior[,1] >.5] = 1  
Pred <- as.factor(Pred)  
confusionMatrix(Pred,test$crim\_median)

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction 0 1  
## 0 3 42  
## 1 63 18  
##   
## Accuracy : 0.1667   
## 95% CI : (0.1062, 0.2434)  
## No Information Rate : 0.5238   
## P-Value [Acc > NIR] : 1.00000   
##   
## Kappa : -0.6443   
##   
## Mcnemar's Test P-Value : 0.05096   
##   
## Sensitivity : 0.04545   
## Specificity : 0.30000   
## Pos Pred Value : 0.06667   
## Neg Pred Value : 0.22222   
## Prevalence : 0.52381   
## Detection Rate : 0.02381   
## Detection Prevalence : 0.35714   
## Balanced Accuracy : 0.17273   
##   
## 'Positive' Class : 0   
##

# Knn

Knn <- knn3(crim\_median~nox + medv, data = train, k = 1)  
summary(Knn)

## Length Class Mode   
## learn 2 -none- list   
## k 1 -none- numeric  
## terms 3 terms call   
## xlevels 0 -none- list   
## theDots 0 -none- list

Pred <- predict(Knn, test, type = "class")  
confusionMatrix(Pred,test$crim\_median)

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction 0 1  
## 0 51 17  
## 1 15 43  
##   
## Accuracy : 0.746   
## 95% CI : (0.6608, 0.8193)  
## No Information Rate : 0.5238   
## P-Value [Acc > NIR] : 2.542e-07   
##   
## Kappa : 0.4901   
##   
## Mcnemar's Test P-Value : 0.8597   
##   
## Sensitivity : 0.7727   
## Specificity : 0.7167   
## Pos Pred Value : 0.7500   
## Neg Pred Value : 0.7414   
## Prevalence : 0.5238   
## Detection Rate : 0.4048   
## Detection Prevalence : 0.5397   
## Balanced Accuracy : 0.7447   
##   
## 'Positive' Class : 0   
##

Knn <- knn3(crim\_median~nox + medv, data = train, k = 2)  
summary(Knn)

## Length Class Mode   
## learn 2 -none- list   
## k 1 -none- numeric  
## terms 3 terms call   
## xlevels 0 -none- list   
## theDots 0 -none- list

Pred <- predict(Knn, test, type = "class")  
confusionMatrix(Pred,test$crim\_median)

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction 0 1  
## 0 46 18  
## 1 20 42  
##   
## Accuracy : 0.6984   
## 95% CI : (0.6103, 0.7769)  
## No Information Rate : 0.5238   
## P-Value [Acc > NIR] : 4.962e-05   
##   
## Kappa : 0.3964   
##   
## Mcnemar's Test P-Value : 0.8711   
##   
## Sensitivity : 0.6970   
## Specificity : 0.7000   
## Pos Pred Value : 0.7188   
## Neg Pred Value : 0.6774   
## Prevalence : 0.5238   
## Detection Rate : 0.3651   
## Detection Prevalence : 0.5079   
## Balanced Accuracy : 0.6985   
##   
## 'Positive' Class : 0   
##

Knn <- knn3(crim\_median~nox + medv, data = train, k = 3)  
summary(Knn)

## Length Class Mode   
## learn 2 -none- list   
## k 1 -none- numeric  
## terms 3 terms call   
## xlevels 0 -none- list   
## theDots 0 -none- list

Pred <- predict(Knn, test, type = "class")  
confusionMatrix(Pred,test$crim\_median)

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction 0 1  
## 0 52 19  
## 1 14 41  
##   
## Accuracy : 0.7381   
## 95% CI : (0.6523, 0.8124)  
## No Information Rate : 0.5238   
## P-Value [Acc > NIR] : 6.741e-07   
##   
## Kappa : 0.473   
##   
## Mcnemar's Test P-Value : 0.4862   
##   
## Sensitivity : 0.7879   
## Specificity : 0.6833   
## Pos Pred Value : 0.7324   
## Neg Pred Value : 0.7455   
## Prevalence : 0.5238   
## Detection Rate : 0.4127   
## Detection Prevalence : 0.5635   
## Balanced Accuracy : 0.7356   
##   
## 'Positive' Class : 0   
##

# Knn with k = 1 is the best

# Findings from each model varies, the best model was by far logistic regression as the misclassification rate is the lowest by far with an accuracy of 92%. This accuracy is scary accurate and could make this model usable in real life.