Minería de Datos - Práctica Calificada N.2

HUERTAS, ANTHONY[1,*]

[1] *Maestría en Estadística, Escuela de Posgrado, Pontificia Universidad Católica del Perú, Lima, Perú*
[*] *Cod: 20173728*

Compiled July 5, 2018

Profesor: Luis Benites

PREGUNTA 1

Esta pregunta debe responderse utilizando el conjunto de datos Weekly, que es parte del paquete ISRL. Esta información es similar en naturaleza a los datos de Smarket que ha sido analizado en http://bit.ly/2JUDp7e, excepto que contiene 1089 declaraciones semanales durante 21 años, desde el comiento de 1990 hasta el final de 2010.

a) Produzca algunos resúmenes numéricos y gráficos de los datos semanales. ¿Parece que hay algún patrón?

Se revisa las variables presentes en la base Weekly

Listing 1. Datos Weekly

```
library(ISLR)
library(ggplot2)
library(knitr)
data("Weekly")
kable(head(Weekly),3)
```

	Year	Lag1	Lag2	Lag3	Lag4	Lag5	Volume	Today	Direction
1	1990	0.816	1.572	-3.936	-0.229	-3.484	0.1549760	-0.270	Down
2	1990	-0.270	0.816	1.572	-3.936	-0.229	0.1485740	-2.576	Down
3	1990	-2.576	-0.270	0.816	1.572	-3.936	0.1598375	3.514	Up

Fig. S1. Datos Weekly.

Se detallan, a continuación, ciertos resúmenes descriptivos de las variables

Listing 2. Weekly: Resumen Descriptivo

```
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  
```

Fig. S2. Datos Weekly: Resumen Descriptivo.

Se analizará la existencia de algún patrón entre las variables continuas, para ello se realiza el cálculo de las correlaciones entre dichas variables, posterioremente la gráfica adecuada entre las que presentan mayor correlación.

Listing 3. Weekly: Matriz de correlación

```
cor_matrix <- round(cor(Weekly[,-9], method = "pearson" ), digits = 4)
cor_matrix[upper.tri(cor_matrix, diag = TRUE)] <- ""
as.data.frame(cor_matrix)
```

	Year	Lag1	Lag2	Lag3	Lag4	Lag5	Volume	Today
Year								
Lag1	-0.0323							
Lag2	-0.0334	-0.0749						
Lag3	-0.03	0.0586	-0.0757					
Lag4	-0.0311	-0.0713	0.0584	-0.0754				
Lag5	-0.0305	-0.0082	-0.0725	0.0607	-0.0757			
Volume	0.8419	-0.065	-0.0855	-0.0693	-0.0611	-0.0585		
Today	-0.0325	-0.075	0.0592	-0.0712	-0.0078	0.011	-0.0331	

Fig. S3. Weekly: Matriz de correlación.

Como se observa, hay una fuerte correlación, positiva, entre las variables **Year** - **Volume**, interpretándose esto como que el número de movimientos ha ido incrementándose con el tiempo. Se observa además que entre las variables **lag** - **Today** , se observan casi nulas correlaciones interpretándose esto como la nula correlación entre el valor de mercado de los días previos y el día actual, algo de esperarse dado lo impredecible que puede resultar el movimiento del mercado en días relativamente cercanos.

A continuación, se presente el gráfico entre **Year** - **Volume**, con el objetivo de visualizar la correlación respectiva.

Listing 4. Weekly: Year - Volume.

```
ggplot(data = Weekly, aes(x = Year, y = Volume))
        + geom_point(color = "firebrick")
        + geom_smooth(method = "lm")
        + theme_bw()
```

En efecto, se observa una tendencia creciente a lo largo de los años.

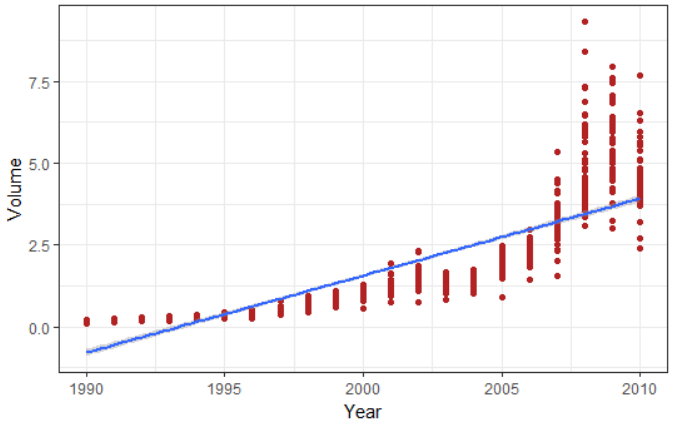

Fig. S4. Datos Weekly: Year - Volume.

Habiéndose previamente observado un patrón de correlacion presente, se presentarán las distribuciones de cada variables **Lag** y **Volume** para cada **Direction**.

Listing 5. Weekly: Histogramas de distribución por cada (Direction).

```
par(mfcol = c(2, 6))
for (i in 2:7) {
        variable <- names(Weekly)[i]
        rango <- seq(min(Weekly[, i]), max(Weekly[, i]), le = 50)
        for (k in 1:2) {
                grupo <- levels(Weekly$Direction)[k]
                x <- Weekly[Weekly$Direction == grupo, variable]
                hist(x, proba = T, col = grey(0.8), main = grupo, xlab = variable)
                lines(rango, dnorm(rango, mean(x), sd(x)), col = "red", lwd = 2)
        }
}
```

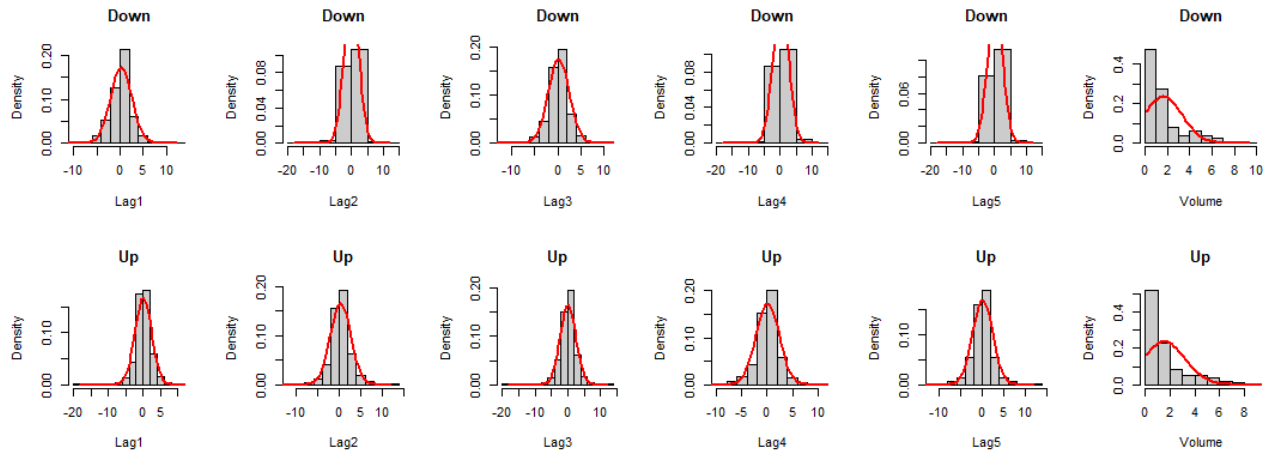

Fig. S5. Weekly: Histogramas de distribución por cada Direction.

Notamos que la mayoría de distribuciones parecen no seguir una distribución normal, esto se verifica mediante el *Test Shapiro-Wilk*

Listing 6. Weekly: Contraste de normalidad Shapiro-Wilk para cada variable en cada dirección.

```
library(reshape2)
library(knitr)
library(dplyr)
Weekly_tidy <- melt(Weekly[,-1], value.name = "valor")
Weekly_tidy %>% group_by(Direction, variable) %>%
        summarise(p_value_Shapiro.test =
                  shapiro.test(valor)$p.value)
```

Los p-valores obtenidos son significativos por lo que se concluye que los datos, respecto a cada variable, no provienen de una distribución normal.

```
# A tibble: 14 x 3
# Groups:   Direction [?]
   Direction variable p_value_Shapiro.test
   <fct>     <fct>                    <dbl>
 1 Down      Lag1                  8.27e- 9
 2 Down      Lag2                  4.92e-14
 3 Down      Lag3                  3.61e- 8
 4 Down      Lag4                  5.70e-15
 5 Down      Lag5                  5.00e-15
 6 Down      Volume                1.73e-24
 7 Down      Today                 2.96e-26
 8 Up        Lag1                  1.96e-16
 9 Up        Lag2                  1.88e-12
10 Up        Lag3                  6.08e-16
11 Up        Lag4                  1.29e-10
12 Up        Lag5                  6.16e-11
13 Up        Volume                1.67e-27
14 Up        Today                 1.07e-25
```

Fig. S6. Weekly: Test Shapiro-Wilk.

b) Use el conjunto de datos completo para realizar una regresión logística con Direction como la respuesta y las cinco variables lag más el Volume como predictores. Use la función **summary** para imprimir los resultados. ¿Aguno de los predictores parece ser estadísticamente significativo? Si es así, ¿cuáles?

Se procede a diseñar un modelo logístico usando todas las variables como predictores; a excepción de **Year** a causa del patrón de correlación presente con la variable **Volume**, y **Today** a causa de que dicha variable es representada por la variable **Direction** en forma binaria.

Listing 7. Weekly: Regresión Logística Completa.

```
# Codificacion de la variable respuesta.
contrasts(Weekly$Direction)

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

Primero se analizan los valores que contrastan a la variable Direction.

Down	0
Up	1

```
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
```

Fig. S7. Weekly: Regresión Logística Completa / Coeficientes estimados.

Según lo obtenido, se observa que la única variable, en el modelo, con un p-valor significativo es **Lag2**. Además, dado que el coeficiente estimado respecto a dicha variables es positivo, indica que si el mercado subió hace dos días, es más probable que suba hoy.

c) Calcule e interprete la matriz de confusión de las predicciones correctas obtenido por una regresión logística.

Listing 8. Weekly: Regresión Logística Completa / Matriz de confusión.

```
predicciones <- predict(object = modelo_logistico, type = "response")

prediccion <- data.frame(probabilidad = predicciones, clase = rep(NA,length(predicciones)))
prediccion[prediccion$probabilidad < 0.5,"clase"] <- "Down"
prediccion[prediccion$probabilidad > 0.5,"clase"] <- "Up"

table(clase_predicha = prediccion$clase, clase_real = Weekly$Direction)

paste( "% de acierto:", mean(prediccion$clase == Weekly$Direction))
paste( "% de error:",mean(prediccion$clase != Weekly$Direction))
```

	Clase Real	
Clase predicha	Down	Up
Down	54	48
Up	430	557

El modelo logístico usando todas las variables como predictoras ha sido capaz de predecir correctamente el 56.1% de las observaciones, mejor de lo que cabría esperar por azar (50%). El *training error* es de 43.9%.

d) Ahora ajuste el modelo de regresión logística utilizando un período de datos de capacitación de 1990 a 2008, con Lag2 como el único predictor. Calcule la matriz de confusión de las predicciones correctas para los datos retenidos (es decir, los datos de 2009 y 2010).

Listing 9. Weekly: Regresión Logística (Modelo 2).

```
train_data <- Weekly[Weekly$Year < 2009,]
test_data <- Weekly[!(Weekly$Year < 2009),]

# Modelo de regresion logistica
modelo <- glm(Direction ~ Lag2, data = train_data, family = "binomial")
```

```
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
```

Fig. S8. Weekly: Regresión Logística (Modelo 2).

Habiéndose diseñado un modelo de regresión logística con datos de entrenamiento y datos de evaluación, se procede a crear la matriz de confusión para la determinación del *test error*.

Listing 10. Weekly: Regresión Logística (Modelo 2) / Matriz de confusión.

```
predicciones2 <- predict(object = modelo, newdata = test_data, type = "response")

predicciones2[predicciones2 > 0.5] <- "Up"
predicciones2[predicciones2 != "Up"] <- "Down"

table(clase_predicha = predicciones2, clase_real = test_data$Direction)

paste("% de acierto:", mean(predicciones2 == test_data$Direction))
paste("% de error:", mean(predicciones2 != test_data$Direction))
```

	Clase Real	
Clase predicha	Down	Up
Down	9	5
Up	34	56

El modelo logístico ha sido capaz de predecir correctamente el 62.5% de las observaciones, mejor de lo que cabría esperar por azar (50%). El *test error* es de 37.5%. Es clara la mejora con respecto al Modelo de Regresión Logístico Completo.

e) Repetir (d) usando LDA

Listing 11. Weekly: Modelo LDA.

```
train_data <- Weekly[Weekly$Year < 2009,]
test_data <- Weekly[!(Weekly$Year < 2009),]

# Modelo LDA
library(MASS)
modelo_lda <- lda(Direction ~ Lag2, data = train_data)
modelo_lda
```

```
Prior probabilities of groups:
     Down        Up
0.4477157 0.5522843

Group means:
            Lag2
Down -0.03568254
Up    0.26036581

Coefficients of linear discriminants:
           LD1
Lag2 0.4414162
```

Fig. S9. Weekly: Modelo LDA.

Habiéndose diseñado un modelo LDA con datos de entrenamiento y datos de evaluación, se procede crear la matriz de confusión para la determinación del *test error*.

Listing 12. Weekly: Modelo LDA / Matriz de confusión.

```
predicciones_lda <- predict(object = modelo_lda, test_data)

table(clase_predicha = predicciones_lda$class,clase_real =  test_data$Direction)
paste("% de acierto:", mean(predicciones_lda$class == test_data$Direction))
paste("% de error:", mean(predicciones_lda$class != test_data$Direction))
```

	Clase Real	
Clase predicha	Down	Up
Down	9	5
Up	34	56

El modelo LDA ha sido capaz de predecir correctamente el 62.5% de las observaciones, mejor de lo que cabría esperar por azar (50%). El *test error* es de 37.5%. Notamos que los resultados son similares que lo obtenido por el modelo 2 de regresión logística.

f) Repetir (d) usando QDA

Listing 13. Weekly: Modelo QDA.

```
modelo_qda <- qda(Direction ~ Lag2, data = train_data)
modelo_qda
```

```
Prior probabilities of groups:
    Down       Up
0.4477157 0.5522843

Group means:
             Lag2
Down -0.03568254
Up    0.26036581
```

Fig. S10. Weekly: Modelo QDA.

Habiéndose diseñado un modelo QDA con datos de entrenamiento y datos de evaluación, se procede crear la matriz de confusión para la determinación del *test error*.

Listing 14. Weekly: Modelo QDA / Matriz de confusión.

```
predicciones_qda <- predict(object = modelo_qda, test_data)

table(clase_predicha = predicciones_qda$class, clase_real =  test_data$Direction)
paste("% de acierto:", mean(predicciones_qda$class == test_data$Direction))
paste("% de error:", mean(predicciones_qda$class != test_data$Direction))
```

	Clase Real	
Clase predicha	Down	Up
Down	0	0
Up	43	61

El modelo QDA ha sido capaz de predecir correctamente el 58.7% de las observaciones, mejor de lo que cabría esperar por azar (50%). El *test error* es de 41.3%. Está predicción trae un *test error* mayor que las obtenidas anteriormente.

g) Repita (d) usando KNN con K=1

Listing 15. Weekly: Modelo KNN, K=1.

```
library(class)
set.seed(604)

prediccion_knn2 <- knn(train = matrix(train_data[, "Lag2"]),
                       test = matrix(test_data[, "Lag2"]) ,
                       cl = train_data[,"Direction"], k = 1 )
```

Habiéndose diseñado un modelo KNN, con k=1, con datos de entrenamiento y datos de evalución, se procede crear la matriz de confusión para la determinación del *test error*.

Listing 16. Weekly: Modelo KNN, K=1 / Matriz de confusión.

```
table(clase_predicha = prediccion_knn2, clase_real = test_data$Direction)

paste("% de acierto:", mean(prediccion_knn2 == test_data$Direction))
paste("% de error:", mean(prediccion_knn2 != test_data$Direction))
```

	Clase Real	
Clase predicha	Down	Up
Down	21	29
Up	22	32

El modelo ha sido capaz de predecir correctamente el 51% de las observaciones, mejor de lo que cabría esperar por azar (50%). El *test error* es de 49%. Está predicción trae un *test error* aún mayor que las obtenidas por el modelo logístico y el modelo LDA.

h) ¿Cúal de estos métodos parece proporcionar los mejores resultados en esta información?

Como pudimos observar, el modelo 2 de regresión logística y el modelo LDA, son las que menor *test error* traen consigo, por lo que se puede decir que proporcionan mejores resultados al recoger una adecuada relación entre el predictor **Lag2** y la variable **Direction**.

i) Experimente con diferentes combinaciones de predictores, incluidas posibles transformaciones e interacciones, para cada unos de los métodos. Informe las variables, el método y la matriz de confusión asociada que parece proporcionar los mejores resultados en los datos extendidos. Tenga en cuenta que también debe experimentar con valores para K en el clasificador KNN.

En **Fig. S7**, si bien se observó que era significativa solo la variable **Lag2**(p-valor = 0.03), también podemos darnos cuenta que la variable **Lag1** es la que menor grado de significancia presenta respecto a las demás variables no significativas (p-valor = 0.1181). Por lo que el análisis será hecha particularmente con esta combinación de predictores.

- Variables: **Lag1**, **Lag2**. Modelo: Logístico.

Listing 17. Weekly: Regresión Logística (Predictores: Lag1, Lag2).

```
modelo <- glm(Direction ~ Lag1 + Lag2, data = train_data, family = "binomial")

predicciones2 <- predict(object = modelo, newdata = test_data, type = "response")

# Se considera como threshold de clasificacion el 0.5
predicciones2[predicciones2 > 0.5] <- "Up"
predicciones2[predicciones2 != "Up"] <- "Down"

# Matriz de confusion
table(clase_predicha = predicciones2, clase_real = test_data$Direction)
paste("% de acierto:", mean(predicciones2 == test_data$Direction))
paste("% de error:", mean(predicciones2 != test_data$Direction))
```

	Clase Real	
Clase predicha	Down	Up
Down	7	8
Up	36	53

El modelo ha sido capaz de predecir correctamente el 57.7% de las observaciones, mejor de lo que cabría esperar por azar (50%). El *test error* es de 42.3%.

- Variables: **Lag1**, **Lag2**. Modelo: LDA.

Listing 18. Weekly: Modelo LDA (Predictores: Lag1, Lag2).

```
modelo_lda <- lda(Direction ~ Lag1 + Lag2, data = train_data)

predicciones_lda <- predict(object = modelo_lda, test_data)

# Matriz de confusion
table(clase_predicha = predicciones_lda$class,clase_real =  test_data$Direction)
paste("% de acierto:", mean(predicciones_lda$class == test_data$Direction))
paste("% de error:", mean(predicciones_lda$class != test_data$Direction))
```

	Clase Real	
Clase predicha	Down	Up
Down	7	8
Up	36	53

El modelo ha sido capaz de predecir correctamente el 57.7% de las observaciones, mejor de lo que cabría esperar por azar (50%). El *test error* es de 42.3%.

– Variables: **Lag1**, **Lag2**. Modelo: LQA.

Listing 19. Weekly: Modelo LQA (Predictoras: Lag1, Lag2).

```
modelo_qda <- qda(Direction ~ Lag1 + Lag2, data = train_data)

predicciones_qda <- predict(object = modelo_qda, test_data)

# Matriz de confusion
table(clase_predicha = predicciones_qda$class,clase_real =  test_data$Direction)
paste("% de acierto:", mean(predicciones_qda$class == test_data$Direction))
paste("% de error:", mean(predicciones_qda$class != test_data$Direction))
```

	Clase Real	
Clase predicha	Down	Up
Down	7	10
Up	36	51

El modelo ha sido capaz de predecir correctamente el 55.8% de las observaciones, mejor de lo que cabría esperar por azar (50%). El *test error* es de 44.2%.

– Variables: **Lag1**, **Lag2**. Modelo: KNN, K=3.

Listing 20. Weekly: Modelo KNN, K=3 (Predictoras: Lag1, Lag2).

```
set.seed(604)
prediccion_knn2 <- knn(train = train_data[,c("Lag1", "Lag2")],
                       test = test_data[,c("Lag1", "Lag2")] ,
                       cl = train_data[,"Direction"], k = 3 )

# Matriz de confusion
table(clase_predicha = prediccion_knn2, clase_real = test_data$Direction)
paste("% de acierto:", mean(prediccion_knn2 == test_data$Direction))
paste("% de error:", mean(prediccion_knn2 != test_data$Direction))
```

	Clase Real	
Clase predicha	Down	Up
Down	22	29
Up	21	32

El modelo ha sido capaz de predecir correctamente el 51.9% de las observaciones, mejor de lo que cabría esperar por azar (50%). El *test error* es de 48.1%.

– Variables: **Lag1**, **Lag2**. Modelo: KNN, K=6.

Listing 21. Weekly: Modelo KNN, K=6 (Predictoras: Lag1, Lag2).

```
set.seed(604)
prediccion_knn2 <- knn(train = train_data[,c("Lag1", "Lag2")],
                       test = test_data[,c("Lag1", "Lag2")] ,
                       cl = train_data[,"Direction"], k = 6 )

# Matriz de confusion
table(clase_predicha = prediccion_knn2, clase_real = test_data$Direction)
paste("% de acierto:", mean(prediccion_knn2 == test_data$Direction))
paste("% de error:", mean(prediccion_knn2 != test_data$Direction))
```

	Clase Real	
Clase predicha	Down	Up
Down	23	29
Up	20	32

El modelo ha sido capaz de predecir correctamente el 52.9% de las observaciones, mejor de lo que cabría esperar por azar (50%). El *test error* es de 47.1%.

– Variables: **Lag1**, **Lag2**, **Lag1**×**Lag2**(Interacción) . Modelo: Logístico.

Listing 22. Weekly: Regresión Logística (Predictoras: Lag1, Lag2, Lag1×Lag2).

```
modelo <- glm(Direction ~ Lag1*Lag2, data = train_data, family = "binomial")

predicciones2 <- predict(object = modelo, newdata = test_data, type = "response")

# Se considera como threshold de clasificacion el 0.5
predicciones2[predicciones2 > 0.5] <- "Up"
predicciones2[predicciones2 != "Up"] <- "Down"

# Matriz de confusion
table(clase_predicha = predicciones2, clase_real = test_data$Direction)
paste("% de acierto:", mean(predicciones2 == test_data$Direction))
paste("% de error:", mean(predicciones2 != test_data$Direction))
```

	Clase Real	
Clase predicha	Down	Up
Down	7	8
Up	36	53

El modelo ha sido capaz de predecir correctamente el 57.7% de las observaciones, mejor de lo que cabría esperar por azar (50%). El *test error* es de 42.3%.

– Variables: **Lag1**, **Lag2**, **Lag1**×**Lag2**(Interacción). Modelo: LDA.

Listing 23. Weekly: Modelo LDA (Predictoras: Lag1, Lag2, Lag1×Lag2).

```
modelo_lda <- lda(Direction ~ Lag1*Lag2, data = train_data)

predicciones_lda <- predict(object = modelo_lda, test_data)

# Matriz de confusion
table(clase_predicha = predicciones_lda$class,clase_real =  test_data$Direction)
paste("% de acierto:", mean(predicciones_lda$class == test_data$Direction))
paste("% de error:", mean(predicciones_lda$class != test_data$Direction))
```

	Clase Real	
Clase predicha	Down	Up
Down	7	8
Up	36	53

El modelo ha sido capaz de predecir correctamente el 57.7% de las observaciones, mejor de lo que cabría esperar por azar (50%). El *test error* es de 42.3%.

– Variables: **Lag1**, **Lag2**, **Lag1**×**Lag2**(Interacción). Modelo: LQA.

Listing 24. Weekly: Modelo LQA (Predictoras: Lag1, Lag2, Lag1×Lag2).

```
modelo_qda <- qda(Direction ~ Lag1*Lag2, data = train_data)

predicciones_qda <- predict(object = modelo_qda, test_data)

# Matriz de confusion
table(clase_predicha = predicciones_qda$class,clase_real =  test_data$Direction)
paste("% de acierto:", mean(predicciones_qda$class == test_data$Direction))
paste("% de error:", mean(predicciones_qda$class != test_data$Direction))
```

	Clase Real	
Clase predicha	Down	Up
Down	23	36
Up	20	25

El modelo ha sido capaz de predecir correctamente el 46.2% de las observaciones, no es un buen porcentaje dado que es menor que el 50%. El *test error* es de 53.8%.

– Variables: **Lag1**, **Lag2**, **Lag1**×**Lag2**(Interacción). Modelo: KNN, K=3.

Listing 25. Weekly: Modelo KNN, K=3 (Predictoras: Lag1, Lag2, Lag1×Lag2).

```
set.seed(604)
train_data$Lag1Lag2 <- train_data$Lag1*train_data$Lag2
test_data$Lag1Lag2 <- test_data$Lag1*test_data$Lag2

prediccion_knn2 <- knn(train = train_data[,c("Lag1", "Lag2","Lag1Lag2")],
                       test = test_data[,c("Lag1", "Lag2","Lag1Lag2")] ,
                       cl = train_data[,"Direction"], k = 3 )

# Matriz de confusion
table(clase_predicha = prediccion_knn2, clase_real = test_data$Direction)
paste("% de acierto:", mean(prediccion_knn2 == test_data$Direction))
paste("% de error:", mean(prediccion_knn2 != test_data$Direction))
```

	Clase Real	
Clase predicha	Down	Up
Down	23	33
Up	20	28

El modelo ha sido capaz de predecir correctamente el 49% de las observaciones, no es un buen porcentaje dado que es menor que el 50%. El *test error* es de 51%.

– Variables: **Lag1**, **Lag2**, **Lag1**×**Lag2**(Interacción). Modelo: KNN, K=6.

Listing 26. Weekly: Modelo KNN, K=6 (Predictoras: Lag1, Lag2, Lag1×Lag2).

```
set.seed(604)
train_data$Lag1Lag2 <- train_data$Lag1*train_data$Lag2
test_data$Lag1Lag2 <- test_data$Lag1*test_data$Lag2

prediccion_knn2 <- knn(train = train_data[,c("Lag1", "Lag2","Lag1Lag2")],
test = test_data[,c("Lag1", "Lag2","Lag1Lag2")] ,
cl = train_data[,"Direction"], k = 6 )

# Matriz de confusion
table(clase_predicha = prediccion_knn2, clase_real = test_data$Direction)
paste("% de acierto:", mean(prediccion_knn2 == test_data$Direction))
paste("% de error:", mean(prediccion_knn2 != test_data$Direction))
```

	Clase Real	
Clase predicha	Down	Up
Down	24	22
Up	19	39

El modelo ha sido capaz de predecir correctamente el 60.6% de las observaciones, mejor de lo que cabría esperar por azar (50%). El *test error* es de 39.4%.

– Variables: **Lag1**, **Lag2**, **Lag3**, **Lag4**, **Lag5**, **log(Volume)** . Modelo: Logístico.

Se aplicó el logartimo a la variable **Volume** para otorgarle un comportamiento semejante al de una distribución normal, luego ha sido representado dicho como variable predictora en un modelo logístico.

Listing 27. Weekly: Regresión Logística (Predictoras: Lag1, Lag2, Lag3, Lag4, Lag5, log(Volume).

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

predicciones2 <- predict(object = modelo, newdata = test_data, type = "response")

# Se considera como threshold de clasificacion el 0.5
predicciones2[predicciones2 > 0.5] <- "Up"
predicciones2[predicciones2 != "Up"] <- "Down"

# Matriz de confusion
table(clase_predicha = predicciones2, clase_real = test_data$Direction)
paste("% de acierto:", mean(predicciones2 == test_data$Direction))
paste("% de error:", mean(predicciones2 != test_data$Direction))
```

	Clase Real	
Clase predicha	Down	Up
Down	17	16
Up	26	45

El modelo ha sido capaz de predecir correctamente el 59.6% de las observaciones, mejor de lo que cabría esperar por azar (50%). El *test error* es de 40.4%.

Se logra observar que de los modelos evaluados, el que presenta menor *test error* es el modelo KNN con K=6 sobre las variables Lag1 y Lag2 en adición de su interacción.

PREGUNTA 2

En este problema, desarrolle un modelo para predecir si un automóvil dado obtiene un millaje de gasolina alto o bajo en función del conjunto de datos Auto del paquete ISRL.

a) Cree una variable binaria **mpg1**, que contenga un 1 si **mpg** contiene un valor por encima de su mediana, y un 0 si **mpg** contiene un valor por debajo de su mediana. Puede calcular la mediana utilizando median(). Tenga en cuenta que puede resultarle útil utilizar la función data.frame() para crear un único conjunto de datos que contenga tanto **mpg1** como las otras variables de Auto.

Listing 28. Datos Auto, con adición de variable binaria.

```
Auto <- data.frame(Auto)
Auto$mpg1 <- rep(0,dim(Auto)[1])

for ( i in 1:dim(Auto)[1]) {
        if (Auto$mpg[i] > median(Auto$mpg) ){ Auto$mpg1[i] = 1}
}

Auto$mpg1 <- factor(Auto$mpg1)
head(Auto)
```

	mpg	cylinders	displacement	horsepower	weight	acceleration	year	origin	name	mpg1
1	18	8	307.0	130	3504	12.0	70	1	chevrolet chevelle malibu	0
2	15	8	350.0	165	3693	11.5	70	1	buick skylark 320	0
3	18	8	318.0	150	3436	11.0	70	1	plymouth satellite	0
4	16	8	304.0	150	3433	12.0	70	1	amc rebel sst	0
5	17	8	302.0	140	3449	10.5	70	1	ford torino	0
6	15	8	429.0	198	4341	10.0	70	1	ford galaxie 500	0

Fig. S11. Datos Auto.

b) Explore los datos gráficamente para investigar la asociación entre **mpg1** y las otras características. ¿Cuál de las otras características parece ser más útil para predecir **mpg1**? Los diagramas de dispersión y los diagramas de caja pueden ser herramientas útiles para responder a esta pregunta. Describe tus hallazgos.

Listing 29. Auto: Diagramas de dispersión y diagramas de caja por cada mpg1.

```
library(ggplot2)
library(scatterplot3d)
library(GGally)

p <- ggpairs(Auto[,-9],aes(color = factor(mpg1)))+theme_bw()

for(i in 1:p$nrow){
        for (j in 1:p$ncol) {
                p[i,j] <- p[i,j] + scale_fill_manual(values = c("#00AFBB","#E7B800"))
                        + scale_color_manual(values = c("#00AFBB","#E7B800"))
        }
}

p
```

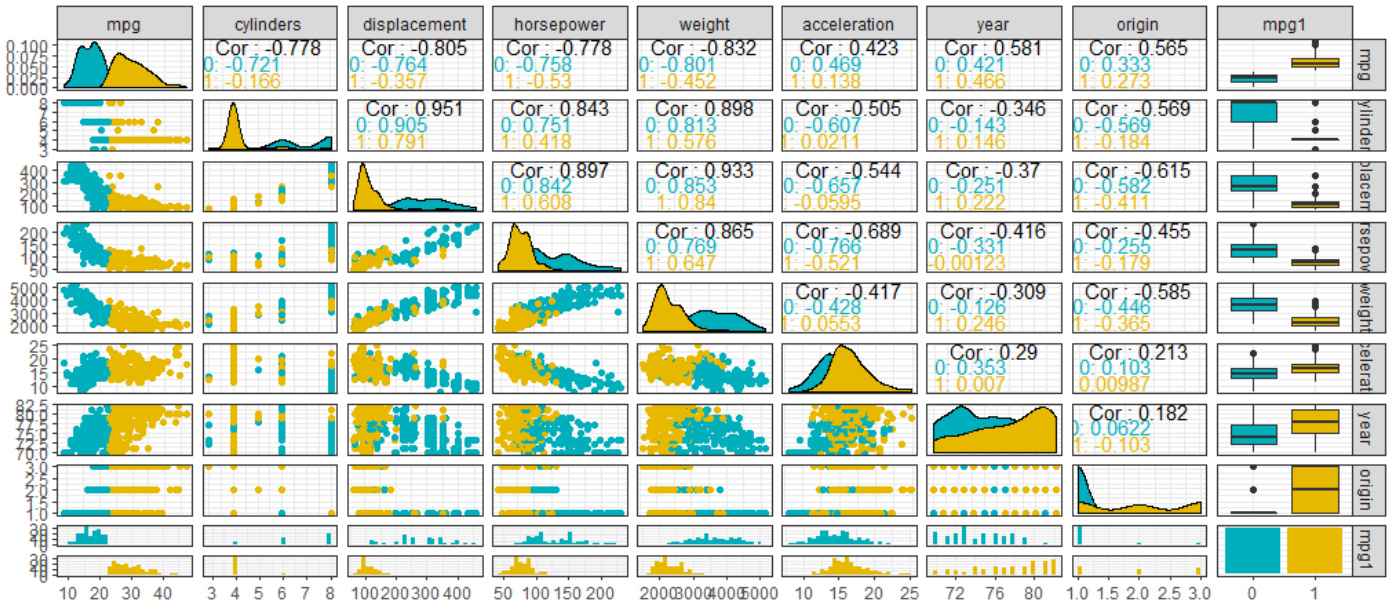

Fig. S12. Auto: Diagramas de dispersión y diagramas de caja.

Habiéndose observado los diagramas de dispersión notamos claros patrones de correlación presentes por ejemplo: las variables **cylinder** y **displacement** presentan altas correlaciones con la variable **weight**; la variable **acceleration** una alta correlación con la variable **horsepower**. Debido a ello, y a las altas correlaciones de **weight**, **horsepower**, **year** y **origin** respecto a la variable mpg, que a su vez está siendo representada por la variable binaria **mpg1**, diremos que son las que mejor parecen estar asociadas a dicha variable.

c) Divida los datos en un conjunto de entrenamiento y un conjunto de prueba.

Se tomará una muestra de 250 datos de la base, que serán tomadas como conjunto de entrenamiento (**train_data**). Lo restante será el conjunto de prueba (**test_data**).

Listing 30. Auto: train_data, test_data.

```
set.seed(604)
m = sample(1:dim(Auto)[1],250)

train_data <- Auto[m,]
test_data <- Auto[-m,]
```

d) Realice LDA en los datos de entrenamiento para predecir **mpg1** usando las variables que parecían más asociadas con **mpg1** en (b). ¿Cuál es el error de prueba (test error) del modelo obtenido?

Listing 31. Auto: Modelo LDA.

```
modelo_lda <- lda(mpg1 ~ horsepower + weight + year + origin, data = train_data)
modelo_lda
```

```
Prior probabilities of groups:
    0     1
0.488 0.512

Group means:
  horsepower   weight     year   origin
0  128.63115 3560.131 74.42623 1.188525
1   78.97656 2338.586 77.78125 1.984375

Coefficients of linear discriminants:
                    LD1
horsepower -0.0009287305
weight     -0.0014219089
year        0.1278130282
origin      0.2629771366
```

Fig. S13. Modelo LDA.

Listing 32. Auto: Modelo LDA / Matriz de confusión.

```
predicciones_lda <- predict(object = modelo_lda, test_data)

table(clase_predicha = predicciones_lda$class,clase_real =  test_data$mpg1)
paste("% de acierto:", mean(predicciones_lda$class == test_data$mpg1))
paste("% de error:", mean(predicciones_lda$class != test_data$mpg1))
```

	Clase Real	
Clase predicha	0	1
0	66	2
1	8	66

El modelo ha sido capaz de predecir correctamente el 93% de las observaciones, mejor de lo que cabría esperar por azar (50%). El *test error* es de 7%.

e) Realice QDA en los datos de entrenamiento para predecir `mpg01` usando las variables que parecían más asociadas con `mpg01` en (b). ¿Cuál es el error de prueba del modelo obtenido?

Listing 33. Auto: Modelo QDA.

```
modelo_qda <- qda(mpg1 ~ horsepower + weight + year + origin, data = train_data)
modelo_qda
```

```
Prior probabilities of groups:
    0     1
0.488 0.512

Group means:
  horsepower   weight     year   origin
0  128.63115 3560.131 74.42623 1.188525
1   78.97656 2338.586 77.78125 1.984375
```

Fig. S14. Auto: Modelo QDA.

Listing 34. Auto: Modelo QDA / Matriz de confusión.

```
predicciones_qda <- predict(object = modelo_qda, test_data)

table(clase_predicha = predicciones_qda$class,clase_real =  test_data$mpg1)
paste("% de acierto:", mean(predicciones_qda$class == test_data$mpg1))
paste("% de error:", mean(predicciones_qda$class != test_data$Direction))
```

	Clase Real	
Clase predicha	0	1
0	67	1
1	7	67

El modelo ha sido capaz de predecir correctamente el 94.4% de las observaciones, mejor de lo que cabría esperar por azar (50%).
El *test error* es de 5.6%.

f) Realice una regresión logística en los datos de entrenamiento para predecir **mpg1** usando las variables que parecían más asociadas con **mpg1** en (b). ¿Cuál es el error de prueba del modelo obtenido?

Listing 35. Auto: Modelo Logístico.

```
modelo <- glm(mpg1 ~ horsepower + weight + year + origin, data = train_data, family = "
    binomial")

summary(modelo)
```

```
Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept) -1.589e+01  5.670e+00  -2.802  0.00507 **
horsepower  -4.611e-02  1.779e-02  -2.591  0.00956 **
weight      -3.821e-03  7.796e-04  -4.902 9.48e-07 ***
year         3.975e-01  8.411e-02   4.726 2.29e-06 ***
origin       5.293e-01  3.255e-01   1.626  0.10392
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Fig. S15. Auto: Modelo Logístico.

Se observa que las variables **horsepower**, **weight**, **year**, son significativos, a excepción de la variable **origin**. Sin embargo, se seguirá optando con el análisis del modelo con dichas variables.
El límite de clasificación se tomo como 0.5.

Listing 36. Auto: Modelo Logístico / Matriz de confusión.

```
predicciones2 <- predict(object = modelo, newdata = test_data, type = "response")

# Se considera como threshold de clasificacion el 0.5
predicciones2[predicciones2 > 0.5] <- 1
predicciones2[predicciones2 <= 0.5] <- 0
predicciones2 <- as.factor(predicciones2)

table(clase_predicha = predicciones2, clase_real = test_data$mpg1)
paste("% de acierto:", mean(predicciones2 == test_data$mpg1))
paste("% de error:", mean(predicciones2 != test_data$mpg1))
```

	Clase Real	
Clase predicha	0	1
0	67	4
1	7	64

El modelo ha sido capaz de predecir correctamente el 92.3% de las observaciones, mejor de lo que cabría esperar por azar (50%).
El *test error* es de 7.7%.

g) Realice KNN en los datos de entrenamiento, con varios valores de K, para predecir mpg01 usando las variables que parecían más asociadas con mpg01 en (b). ¿Cuál es el error de prueba del modelo obtenido?

- Modelo KNN, K=1.

Listing 37. Auto: Modelo KNN, K=1.

```
        set.seed(604)
        prediccion_knn2 <- knn(train = train_data[,c("horsepower", "weight","year","
            origin")],
        test = test_data[,c("horsepower", "weight","year","origin")] ,
        cl = train_data[,"mpg1"], k = 1 )

        # Matriz de confusion
        table(clase_predicha = prediccion_knn2, clase_real = test_data$mpg1)
        paste("% de acierto:", mean(prediccion_knn2 == test_data$mpg1))
        paste("% de error:", mean(prediccion_knn2 != test_data$mpg1))
```

	Clase Real	
Clase predicha	0	1
0	60	9
1	14	59

El modelo ha sido capaz de predecir correctamente el 83.8% de las observaciones, mejor de lo que cabría esperar por azar (50%). El *test error* es de 16.2%.

- Modelo KNN, K=3.

Listing 38. Auto: Modelo KNN, K=3.

```
set.seed(604)
prediccion_knn2 <- knn(train = train_data[,c("horsepower", "weight","year","origin")],
                       test = test_data[,c("horsepower", "weight","year","origin")] ,
                       cl = train_data[,"mpg1"], k = 3 )

# Matriz de confusion
table(clase_predicha = prediccion_knn2, clase_real = test_data$mpg1)
paste("% de acierto:", mean(prediccion_knn2 == test_data$mpg1))
paste("% de error:", mean(prediccion_knn2 != test_data$mpg1))
```

	Clase Real	
Clase predicha	0	1
0	67	10
1	7	58

El modelo ha sido capaz de predecir correctamente el 88% de las observaciones, mejor de lo que cabría esperar por azar (50%). El *test error* es de 12%.

- Modelo KNN, K=6.

Listing 39. Auto: Modelo KNN, K=6.

```
set.seed(604)
prediccion_knn2 <- knn(train = train_data[,c("horsepower", "weight","year","origin")],
                       test = test_data[,c("horsepower", "weight","year","origin")] ,
                       cl = train_data[,"mpg1"], k = 6 )

# Matriz de confusion
table(clase_predicha = prediccion_knn2, clase_real = test_data$mpg1)
paste("% de acierto:", mean(prediccion_knn2 == test_data$mpg1))
paste("% de error:", mean(prediccion_knn2 != test_data$mpg1))
```

	Clase Real	
Clase predicha	0	1
0	66	9
1	8	59

El modelo ha sido capaz de predecir correctamente el 88% de las observaciones, mejor de lo que cabría esperar por azar (50%). El *test error* es de 12%.

PREGUNTA 3

Utilizando el conjunto de datos de Boston del paquete MASS, ajuste los modelos de clasificación para predecir la tasa de delincuencia en los suburbios de Boston que esté por encima o por debajo de la mediana. Explore los modelos de regresión logística, LDA y KNN usando varios subconjuntos de predictores. Describe sus hallazgos.

Se revisan las variables presentes en la base Boston, con adición de una variable adicional **indice** que representa por 1 si la tasa de delincuencia en los suburbios de Boston (**crime**) que esté por encima, y 0 si está debajo de la mediana.

Listing 40. Datos Boston, con adición de la variable binaria

```
Boston$indice <- rep(0,dim(Boston)[1])
for ( i in 1:dim(Boston)[1]) {
        if (Boston$crim[i] > median(Boston$crim) ){ Boston$indice[i] = 1}
}
Boston$indice <- factor(Boston$indice)
head(Boston,3)
```

	crim	zn	indus	chas	nox	rm	age	dis	rad	tax	ptratio	black	lstat	medv	indice
1	0.00632	18.0	2.31	0	0.5380	6.575	65.2	4.0900	1	296	15.3	396.90	4.98	24.0	0
2	0.02731	0.0	7.07	0	0.4690	6.421	78.9	4.9671	2	242	17.8	396.90	9.14	21.6	0
3	0.02729	0.0	7.07	0	0.4690	7.185	61.1	4.9671	2	242	17.8	392.83	4.03	34.7	0

Fig. S16. Datos Boston, con adición de la variable binaria.

Se crean los datos de entrenamiento (**train_data**) y de prueba (**test_data**).

Listing 41. Boston (train_data,test_data).

```
set.seed(604)
m = sample(1:dim(Boston)[1],400)
train_data <- Boston[m,]
test_data <- Boston[-m,]
```

A continuación, se evaluará mediante la función stepAIC cuales variables son significativas en un modelo de regresión logístico.

Listing 42. Boston: Modelo Logístico Completo / StepAIC.

```
library(MASS)
modelo_logistico <- glm(indice ~ zn + indus + chas + nox + rm + age + dis + rad + tax + ptratio +
                        black + lstat + medv, data = train_data, family = "binomial")
stepAIC(modelo_logistico)
```

```
Coefficients:
(Intercept)          zn         nox         dis         rad         tax     ptratio
 -32.367421   -0.077825   45.062247    0.615370    0.719751   -0.007219    0.258134
      black       lstat        medv
  -0.010628    0.093193    0.127506

Degrees of Freedom: 399 Total (i.e. Null);  390 Residual
Null Deviance:      554.5
Residual Deviance: 177  AIC: 197
```

Fig. S17. Boston: Modelo Logístico Completo / StepAIC.

Como observamos el mejor modelo logístico resultaría con uso de las variables zn, nox, dis, rad, tax, ptratio, black, lstat, medv. Estas variables serán usadas como variables predictoras en los modelos diseñados a continuación

- Modelo Logístico.

Listing 43. Boston: Regresión Logística.

```
modelo <- glm(indice ~ zn + nox + dis + rad + tax + ptratio + black + lstat + medv,
              data = train_data, family = "binomial")
summary(modelo)
```

```
Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -32.367421   6.680082  -4.845 1.26e-06 ***
zn           -0.077825   0.033565  -2.319  0.02041 *
nox          45.062247   7.109596   6.338 2.32e-10 ***
dis           0.615370   0.233754   2.633  0.00847 **
rad           0.719751   0.159650   4.508 6.53e-06 ***
tax          -0.007219   0.002728  -2.647  0.00813 **
ptratio       0.258134   0.120172   2.148  0.03171 *
black        -0.010628   0.005827  -1.824  0.06818 .
lstat         0.093193   0.047166   1.976  0.04817 *
medv          0.127506   0.045828   2.782  0.00540 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Fig. S18. Boston: Modelo Logístico.

Listing 44. Boston: Modelo Logístico / Matriz de confusión.

```
predicciones2 <- predict(object = modelo, newdata = test_data, type = "response")

# Se considera como threshold de clasificacion el 0.5
predicciones2[predicciones2 > 0.5] <- 1
predicciones2[predicciones2 <= 0.5] <- 0
predicciones2 <- as.factor(predicciones2)

table(clase_predicha = predicciones2, clase_real = test_data$indice)
paste("% de acierto:", mean(predicciones2 == test_data$indice))
paste("% de error:", mean(predicciones2 != test_data$indice))
```

	Clase Real	
Clase predicha	0	1
0	49	7
1	4	46

El modelo ha sido capaz de predecir correctamente el 89.6% de las observaciones, mejor de lo que cabría esperar por azar (50%). El *test error* es de 10.4%.

- Modelo LDA.

Listing 45. Boston: Modelo LDA.

```
modelo_lda <- lda(indice ~ zn + nox + dis + rad + tax + ptratio + black + lstat + medv,
                  data = train_data)
modelo_lda
```

Listing 46. Boston: Modelo LDA / Matriz de confusión.

```
predicciones_lda <- predict(object = modelo_lda, test_data)

table(clase_predicha = predicciones_lda$class,clase_real =  test_data$indice)
paste("% de acierto:", mean(predicciones_lda$class == test_data$indice))
paste("% de error:", mean(predicciones_lda$class != test_data$indice))
```

	Clase Real	
Clase predicha	0	1
0	50	12
1	3	41

```
Prior probabilities of groups:
  0   1
0.5 0.5

Group means:
       zn       nox      dis   rad    tax ptratio    black    lstat    medv
0 22.9025 0.4711035 5.050753  4.18 304.19 17.9500 388.6413  9.25515 25.1605
1  1.1100 0.6371700 2.515470 14.66 505.51 18.9295 329.0831 15.91580 20.0515

Coefficients of linear discriminants:
                 LD1
zn      -0.0086673059
nox      9.1024825093
dis      0.0103716950
rad      0.0673880393
tax     -0.0003219227
ptratio  0.0372177905
black   -0.0009386438
lstat    0.0291447508
medv     0.0446705998
```

Fig. S19. Boston: Modelo LDA.

El modelo ha sido capaz de predecir correctamente el 85.8% de las observaciones, mejor de lo que cabría esperar por azar (50%). El *test error* es de 14.2%.

- Modelo KNN, K=1.

Listing 47. Boston: Modelo KNN, K=1.

```
set.seed(604)
prediccion_knn2 <- knn(train = train_data[,c("zn","nox","dis","rad","tax","ptratio","black",
                                             "lstat","medv")],
                       test = test_data[,c("zn","nox","dis","rad","tax","ptratio","black",
                                           "lstat","medv")] ,
                       cl = train_data[,"indice"], k = 1 )

table(clase_predicha = prediccion_knn2, clase_real = test_data$indice)
paste("% de acierto:", mean(prediccion_knn2 == test_data$indice))
paste("% de error:", mean(prediccion_knn2 != test_data$indice))
```

	Clase Real	
Clase predicha	0	1
0	46	3
1	7	50

El modelo ha sido capaz de predecir correctamente el 90.6% de las observaciones, mejor de lo que cabría esperar por azar (50%). El *test error* es de 9.4%.

- Modelo KNN, K=3.

Listing 48. Boston: Modelo KNN, K=3.

```
set.seed(604)
prediccion_knn2 <- knn(train = train_data[,c("zn","nox","dis","rad","tax","ptratio","black",
                                             "lstat","medv")],
                       test = test_data[,c("zn","nox","dis","rad","tax","ptratio","black",
                                           "lstat","medv")] ,
                       cl = train_data[,"indice"], k = 3 )

table(clase_predicha = prediccion_knn2, clase_real = test_data$indice)
paste("% de acierto:", mean(prediccion_knn2 == test_data$indice))
paste("% de error:", mean(prediccion_knn2 != test_data$indice))
```

	Clase Real	
Clase predicha	0	1
0	49	4
1	1	49

El modelo ha sido capaz de predecir correctamente el 92.5% de las observaciones, mejor de lo que cabría esperar por azar (50%). El *test error* es de 7.4%.

- Modelo: KNN, K=6.

Listing 49. Boston: Modelo KNN, K=6.

```
set.seed(604)
prediccion_knn2 <- knn(train = train_data[,c("zn","nox","dis","rad","tax","ptratio","black",
                                             "lstat","medv")],
                       test = test_data[,c("zn","nox","dis","rad","tax","ptratio","black",
                                           "lstat","medv")] ,
                       cl = train_data[,"indice"], k = 6 )

table(clase_predicha = prediccion_knn2, clase_real = test_data$indice)
paste("% de acierto:", mean(prediccion_knn2 == test_data$indice))
paste("% de error:", mean(prediccion_knn2 != test_data$indice))
```

	Clase Real	
Clase predicha	0	1
0	47	4
1	6	49

El modelo ha sido capaz de predecir correctamente el 90.6% de las observaciones, mejor de lo que cabría esperar por azar (50%). El *test error* es de 9.4%.

Podemos concluir como mejor modelo, es establecido por KNN con K=3, al presentar menor *test error* que las demás. (7.4%).