

Actividad 3: Modelización Predictiva

2022-12-18

Índice

Introducción	1
1. Regresión Lineal	2
1.1 Modelo de regresión lineal (variables cuantitativas)	2
1.2 Modelo de regresión lineal (variables cuantitativas y cualitativas).	4
1.3 Diagnosis del modelo.	6
1.4 Predicción del modelo	7
2 Regresión logística.	7
2.1 Generación de los conjuntos de entrenamiento y de test	8
2.2 Estimación del modelo con el conjunto de entrenamiento e interpretación	8
2.3 Cálculo de las OR (Odds-Ratio)	11
2.4 Matriz de confusión	12
2.5 Predicción	13
2.6 Bondad del ajuste	13
2.7 Curva ROC	15
3 Informe Ejecutivo	16
3.1 Presentación de los principales resultados del estudio en una tabla	17
3.2 Resumen ejecutivo. Conclusiones del análisis	18

Introducción

En esta actividad se utilizará el dataset **datSat_Air**. Un estudio de satisfacción de clientes de un determinado aeropuerto internacional.

El objetivo principal de este estudio es averiguar cuáles son los factores que más influyen en que un pasajero se muestre satisfecho y cuáles no.

El archivo contiene aproximadamente 129446 registros y 20 variables. Las principales variables son:

- Satisfaction: Satisfacción del viajero, medida en dos categorías: (neutral or dissatisfied y satisfied).
- Gender: Sexo biológico.

- Customer_Type: Tipo de cliente.
- Age: Edad.
- Type_Travel: Tipo de viaje.
- Class: Tipo de tarifa.
- Distance: Distancia entre origen y destino.
- Departure_Delay: Tiempo de retraso en la salida, en minutos.
- Arrival_Delay: Tiempo de retraso en la llegada a destino, en minutos..
- Seat_comfort: Comodidad del asiento.
- Food_drink: Comida y bebida.
- Gate: Distancia a la puerta de embarque.
- Wifi: Servicio de Wifi.
- Ent: Entretenimiento.
- Ease_booking: Fácil booking.
- Service: Servicio en general.
- Baggage_handling: Equipaje de mano.
- Checkin_service: Checking.
- Cleanliness: Limpieza.
- Online_boarding: Embarque online.

1. Regresión Lineal

Guardamos el dataset en la variable data

```
data<-read.csv("datSat_Air.csv")
data$satisfaction_re<-data$satisfaction
```

1.1 Modelo de regresión lineal (variables cuantitativas)

- Estimad por mínimos cuadrados ordinarios un modelo lineal que explique la variable Arrival_Delay en función de la variable Distance.
- Se añadirá al modelo anterior la variable Departure_Delay. ¿Existe una mejora del ajuste?. Razonar.

a)

La variable dependiente es *Arrival_delay* y la independiente es *Distance*. La función **lm()** utilza los mínimos cuadrados ordinarios para medir el error.

Convertimos las variables a numeric

```
data$Arrival_Delay<-as.numeric(data$Arrival_Delay)
data$Distance<-as.numeric(data$Distance)
model1 <- lm(Arrival_Delay ~ Distance, data = data)
summary(model1)
```

```

## 
## Call:
## lm(formula = Arrival_Delay ~ Distance, data = data)
## 
## Residuals:
##    Min     1Q Median     3Q    Max 
## -34.00 -15.30 -11.74  -1.29 477.72 
## 
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept) 7.239e+00 2.166e-01 33.42   <2e-16 ***
## Distance    3.850e-03 9.709e-05 39.66   <2e-16 ***  
## --- 
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1 
## 
## Residual standard error: 35.86 on 129444 degrees of freedom
## Multiple R-squared:  0.012, Adjusted R-squared:  0.012 
## F-statistic:  1573 on 1 and 129444 DF, p-value: < 2.2e-16

```

El valor de p-value es de 2.2e-16. Un threshold típico es 0.05, como el valor de p es inferior lo contamos como estadísticamente significante.

La relación será lineal. A mayor **Distance**, mayor **Arrival_Delay**. Desafortunadamente, el valor de **Adjusted R-squared** no es demasiado alto (0.012) por lo que la correlación no es muy explicativa. Si llamamos la función **cor()** obtenemos 0.1096.

b)

```

data$Departure_Delay<-as.numeric(data$Departure_Delay)
model2 <- lm(Arrival_Delay ~ Distance + Departure_Delay, data = data)
summary(model2)

```

```

## 
## Call:
## lm(formula = Arrival_Delay ~ Distance + Departure_Delay, data = data)
## 
## Residuals:
##    Min     1Q Median     3Q    Max 
## -53.422 -1.951 -0.787 -0.404 236.671 
## 
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept) 6.030e-01 6.083e-02 9.913   < 2e-16 ***
## Distance    9.530e-05 2.733e-05 3.487 0.000488 ***  
## Departure_Delay 9.761e-01 7.905e-04 1234.821   < 2e-16 ***  
## --- 
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1 
## 
## Residual standard error: 10.03 on 129443 degrees of freedom
## Multiple R-squared:  0.9227, Adjusted R-squared:  0.9227 
## F-statistic: 7.724e+05 on 2 and 129443 DF, p-value: < 2.2e-16

```

El valor de p-value sigue indicando ser estadísticamente significante. Con una relación lineal positiva entre la variable dependiente y ambas variables independientes.

En este caso el valor del coeficiente de determinación ajustado es de 0.9227. Por lo que con estas dos variables ya tenemos un modelo predictivo bastante bueno y *sí* existe una mejora del ajuste.

1.2 Modelo de regresión lineal (variables cuantitativas y cualitativas).

- a) Se añadirá al modelo del apartado 1.b), las variables cualitativas ordinales Service, Food_drink y satisfaction, junto con la variable cualitativa nominal Customer_Type. A la vista de los resultados, estudiar si son o no significativas. Decidid cuáles de las variables explicativas propuestas hasta el momento deben quedarse en el modelo de regresión lineal. Se le llamará modelo final ModelF.
- b) Comprobad si existen o no problemas de colinealidad en dicho modelo final ModelF.

a)

```

data$Service<-as.numeric(data$Service)
data$Food_drink<-as.numeric(data$Food_drink)
data$satisfaction<-factor(data$satisfaction)
data$Customer_Type<-factor(data$Customer_Type)
model3 <- lm(Arrival_Delay ~ Distance + Departure_Delay + Service + Food_drink + satisfaction + Customer_Type, data = data)
summary(model3)

##
## Call:
## lm(formula = Arrival_Delay ~ Distance + Departure_Delay + Service +
##     Food_drink + satisfaction + Customer_Type, data = data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max 
## -53.770  -2.186  -0.668  -0.194 237.002 
##
## Coefficients:
##                               Estimate Std. Error t value Pr(>|t|)    
## (Intercept)             1.143e+00  1.239e-01   9.227 < 2e-16 ***
## Distance                8.383e-05  2.733e-05   3.067  0.00216 **  
## Departure_Delay          9.753e-01  7.922e-04 1231.181 < 2e-16 ***
## Service                 -6.457e-02  2.344e-02  -2.755  0.00588 **  
## Food_drink               -2.467e-02  1.945e-02  -1.268  0.20464  
## satisfactionsatisfied    -7.060e-01  6.285e-02  -11.233 < 2e-16 *** 
## Customer_TypeLoyal Customer  2.137e-01  7.537e-02   2.836  0.00457 ** 
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 10.03 on 129439 degrees of freedom
## Multiple R-squared:  0.9228, Adjusted R-squared:  0.9228 
## F-statistic: 2.579e+05 on 6 and 129439 DF,  p-value: < 2.2e-16

```

Las variables Service, Food_drink y satisfaction tienen una relación inversamente proporcional (lineal negativa). Es decir, que a mayor retraso de llegada menores valoraciones en estas variables.

Las variables Distance, Departure_Delay y Customer_Type son positivas (Siendo disloyal=0 y Loyal=1).

El coeficiente *Adjusted R-squared* ha mejorado ligeramente a 0.9228

Mirando el p-value vemos que todas son significativas con excepción de Food_drink. Por tanto, descartamos esa y la variable Customer_Type al ser confusa.

```

data$satisfaction<-as.numeric(data$satisfaction)
modelF <- lm(Arrival_Delay ~ Distance + Departure_Delay + Service + satisfaction, data = data)
summary(modelF)

##
## Call:
## lm(formula = Arrival_Delay ~ Distance + Departure_Delay + Service +
##      satisfaction, data = data)
##
## Residuals:
##    Min     1Q   Median     3Q    Max 
## -53.712 -2.186 -0.643 -0.248 237.063 
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept) 1.895e+00 1.191e-01 15.915 < 2e-16 ***
## Distance    8.301e-05 2.733e-05  3.038 0.00239 **  
## Departure_Delay 9.754e-01 7.921e-04 1231.403 < 2e-16 ***
## Service     -6.577e-02 2.344e-02  -2.806 0.00501 **  
## satisfaction -6.647e-01 5.997e-02 -11.085 < 2e-16 *** 
## ---        
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1 
##
## Residual standard error: 10.03 on 129441 degrees of freedom
## Multiple R-squared:  0.9228, Adjusted R-squared:  0.9228 
## F-statistic: 3.868e+05 on 4 and 129441 DF, p-value: < 2.2e-16

```

b)

Vemos que la correlación entre las variables independientes es de -0.07586, -0.032 y 0.111. Por lo que son muy bajas y es un indicio de que no hay colinealidad.

```
cor(x = data$satisfaction, y = data$Departure_Delay)
```

```
## [1] -0.07586039
```

```
cor(x = data$Distance, y = data$Departure_Delay)
```

```
## [1] 0.1112709
```

```
cor(x = data$Service, y = data$Distance)
```

```
## [1] -0.03233923
```

A continuación, calculamos el FIV(factor de inflación de la varianza). Para detectar problemas en la multicolinalidad.

```
library(car)
```

```
## Loading required package: carData
```

```
vif(modelF)

##          Distance Departure_Delay      Service satisfaction
##        1.013852       1.017933       1.142322       1.147393
```

Finalmente, calculamos el valor FIV con la fórmula $1/(1-R^2)$

```
1/(1-summary(modelF)$r.squared)
```

```
## [1] 12.95221
```

El valor FIV es cercano a 1 para las variables independientes por lo que es menor a 12.95. Por lo que asumimos que no existe multicolinealidad.

Por tanto, mantenemos en el modelF las variables explicativas.

1.3 Diagnosis del modelo.

valores ajustados frente a los residuos

```
rstd<-rstandard(modelF)
adjusted<-fitted(modelF)
plot(adjusted, rstd)
```

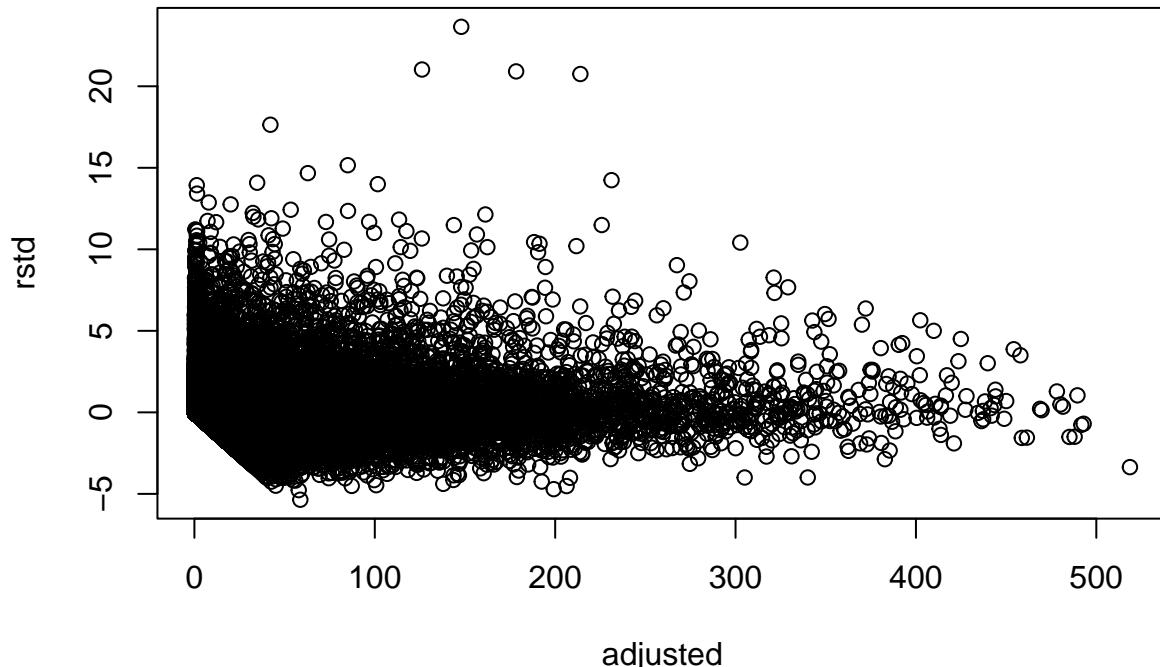
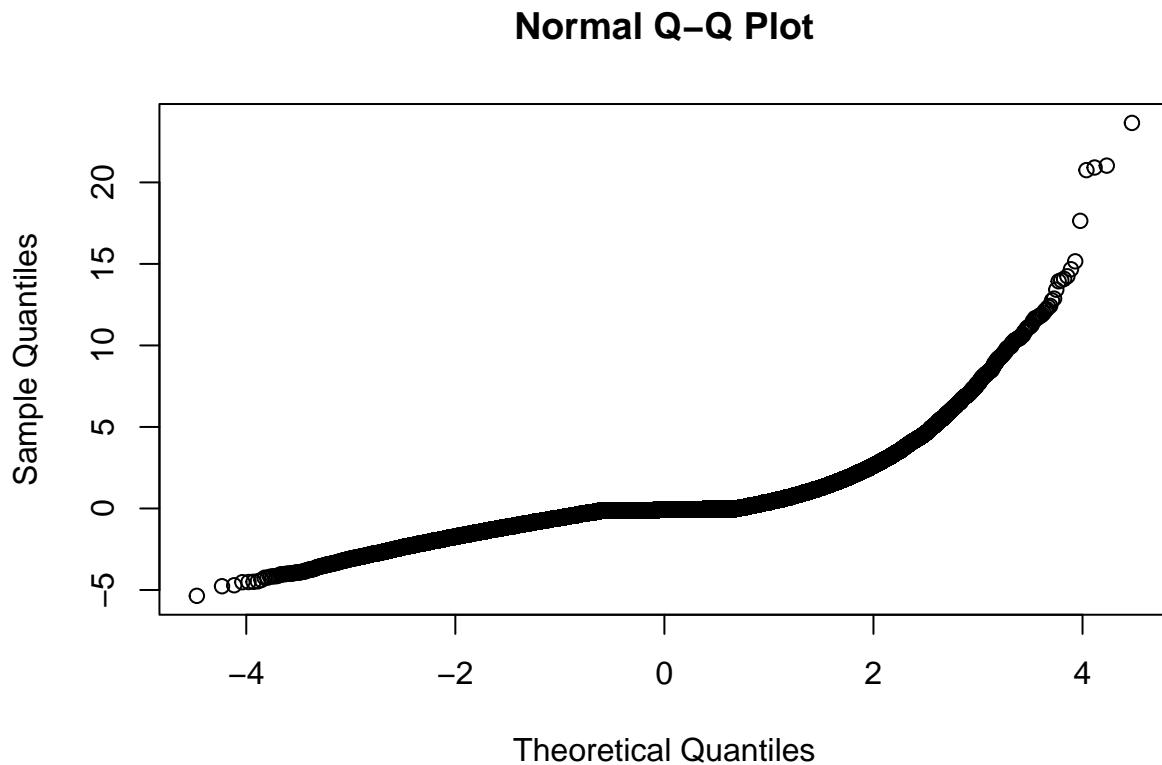


gráfico cuantil-cuantil

```
qqnorm(rstd)
```



El gráfico Normal Q-Q Plot muestra que los datos se ajustan a una normal.

El gráfico de los valores ajustados frente a los residuos muestra un patrón no aleatorio. Pues no suponemos varianza constante en el modelF.

1.4 Predicción del modelo

Según ModelF, calculad el retraso en la llegada del vuelo, si un viajero satisfecho ha recorrido una distancia de 2500 millas y ha tenido un retraso en la salida de 30 minutos. Se conoce que dicho viajero a puntuado con 3, su nivel de satisfacción sobre el servicio (Service).

```
viajero<-data.frame(Distance=2500,Departure_Delay=30,Service=3, satisfaction=2)  
predict(modelF, viajero)
```

```
##          1  
## 29.83705
```

El modelo predice un tiempo de retraso en la llegada del vuelo de 29.84 minutos. (29 minutos y 50 segundos).

2 Regresión logística.

La variable **satisfaction_re** tomará valores 0 ("neutral o dissatisfied") o 1 ("satisfied").

```

data$satisfaction_re[data$satisfaction_re=='neutral or dissatisfied']<-0
data$satisfaction_re[data$satisfaction_re=='satisfied']<-1
data$satisfaction_re<-as.numeric(data$satisfaction_re)

```

2.1 Generación de los conjuntos de entrenamiento y de test

Generad los conjuntos de datos para entrenar el modelo (training) y para testarlo (testing). Se puede fijar el tamaño de la muestra de entrenamiento a un 80% del original.

```

set.seed(1) # Usamos una seed para reproducir el ejemplo
sample <- sample(c(TRUE, FALSE), nrow(data), replace=TRUE, prob=c(0.8,0.2))
train <- data[sample, ]
test <- data[!sample, ]

```

Comprobación:

```
cat("Tamaño de entrenamiento es: ", nrow(train)/nrow(data))
```

```
## Tamaño de entrenamiento es: 0.8006505
```

2.2 Estimación del modelo con el conjunto de entrenamiento e interpretación

- Estimad el modelo de regresión logística siendo la variable dependiente satisfaction_re y tomando todas las variables explicativas de la base de datos. Tened en cuenta la variable satisfaction sin recodificar no puede ser una variable explicativa

```

# Preprocesamos algunas variables
data$Gender<-factor(data$Gender)
data>Type_Travel<-factor(data>Type_Travel)
data$Class<-factor(data$Class)
data$Age<-as.numeric(data$Age)
data$Seat_comfort<-as.numeric(data$Seat_comfort)
data$Gate<-as.numeric(data$Gate)
data$Wifi<-as.numeric(data$Wifi)
data$Ent<-as.numeric(data$Ent)
data$Ease_booking<-as.numeric(data$Ease_booking)
data$Baggage_handling<-as.numeric(data$Baggage_handling)
data$Checkin_service<-as.numeric(data$Checkin_service)
data$Cleanliness<-as.numeric(data$Cleanliness)

```

Creamos el modelo. Indicamos la *family* a la que queremos añadir el modelo. Por defecto, está es gaussian como indica la documentación. La corregimos a binomial para que sea una función logística.

```
model4<-glm(satisfaction_re~satisfaction+Gender+Customer_Type+Age+Type_Travel+Class+Distance+Seat_comfo
```

```
## Warning: glm.fit: algorithm did not converge
```

```

summary(model4)

##
## Call:
## glm(formula = satisfaction_re ~ satisfaction + Gender + Customer_Type +
##      Age + Type_Travel + Class + Distance + Seat_comfort + Food_drink +
##      Gate + Wifi + Ent + Ease_booking + Service + Baggage_handling +
##      Checkin_service + Cleanliness + Online_boarding + Departure_Delay +
##      Arrival_Delay, family = binomial, data = data)
##
## Deviance Residuals:
##      Min        1Q     Median        3Q       Max
## -2.409e-06 -2.409e-06  2.409e-06  2.409e-06  2.409e-06
##
## Coefficients:
##                               Estimate Std. Error z value Pr(>|z|)
## (Intercept)                -7.970e+01  7.657e+03 -0.010  0.992
## satisfaction               5.313e+01  2.743e+03  0.019  0.985
## GenderMale                 9.088e-11  2.068e+03  0.000  1.000
## Customer_TypeLoyal Customer 2.591e-10  3.223e+03  0.000  1.000
## Age                        -3.308e-13 7.239e+01  0.000  1.000
## Type_TravelPersonal Travel -4.923e-10 3.001e+03  0.000  1.000
## ClassEco                   -9.054e-11 2.816e+03  0.000  1.000
## ClassEco Plus              -1.228e-10 4.277e+03  0.000  1.000
## Distance                    4.457e-13 1.040e+00  0.000  1.000
## Seat_comfort                2.507e-10 1.113e+03  0.000  1.000
## Food_drink                  -2.013e-10 1.109e+03  0.000  1.000
## Gate                         1.779e-10 9.295e+02  0.000  1.000
## Wifi                         2.894e-10 1.058e+03  0.000  1.000
## Ent                          2.888e-11 1.030e+03  0.000  1.000
## Ease_booking                 -2.761e-10 1.424e+03  0.000  1.000
## Service                      5.221e-12 1.044e+03  0.000  1.000
## Baggage_handling             -6.350e-11 1.184e+03  0.000  1.000
## Checkin_service              9.753e-11 8.684e+02  0.000  1.000
## Cleanliness                  1.120e-10 1.225e+03  0.000  1.000
## Online_boarding              1.937e-11 1.209e+03  0.000  1.000
## Departure_Delay              -2.222e-11 1.003e+02  0.000  1.000
## Arrival_Delay                2.524e-11 9.879e+01  0.000  1.000
##
## (Dispersion parameter for binomial family taken to be 1)
##
## Null deviance: 1.7828e+05 on 129445 degrees of freedom
## Residual deviance: 7.5099e-07 on 129424 degrees of freedom
## AIC: 44
##
## Number of Fisher Scoring iterations: 25

```

- b) Estudiad la presencia o no de colinealidad. En el caso de existir, eliminar la variable o variables que consideréis.

```
vif(model4)
```

```
##          GVIF Df GVIF^(1/(2*Df))
```

```

## satisfaction      1.902144  1      1.379182
## Gender           1.090811  1      1.044419
## Customer_Type   1.586261  1      1.259468
## Age              1.222516  1      1.105674
## Type_Travel     1.963667  1      1.401309
## Class            1.930933  2      1.178805
## Distance         1.163401  1      1.078611
## Seat_comfort    2.453197  1      1.566269
## Food_drink       2.617283  1      1.617802
## Gate             1.503847  1      1.226314
## Wifi              1.985686  1      1.409144
## Ent               1.962517  1      1.400899
## Ease_booking     3.527778  1      1.878238
## Service          1.797753  1      1.340803
## Baggage_handling 1.913531  1      1.383304
## Checkin_service  1.222924  1      1.105859
## Cleanliness      2.031511  1      1.425311
## Online_boarding  2.513739  1      1.585478
## Departure_Delay 12.947396  1      3.598249
## Arrival_Delay    12.966181  1      3.600858

```

Vemos una alta colinealidad en Departure_Delay y Arrival_Delay. Esto es lógico, ya que si un vuelo sale con retraso, lo más probable es que llegue al destino con retraso. Debemos quitar 1 de las 2 variables para simplificar el modelo. Si llamamos la misma función *vif()* con un modelo sin una de estas variables no encontramos problemas de multicolinealidad.

He optado por quitar la variable *Arrival_Delay* y mantener *Departure_Delay*.

Además hemos de quitar la variables *satisfaction* ya que es identica a *satisfaction_re*

```
cor(x = data$satisfaction, y = data$satisfaction_re)
```

```
## [1] 1
```

c) Una vez corregido el modelo por la presencia o no de colinealidad, se pide:

- Interpretad la salida del modelo final. Se le llamará ModlgF.
- Resumid cuáles de las variables pueden considerarse factores de riesgo o protección.

```
modlgF<-glm(satisfaction_re~Gender+Customer_Type+Age+Type_Travel+Class+Distance+Seat_comfort+Food_drink)
```

OR < 1 es un factor de protección. (Menos probable cuando es presente)

OR > 1 es un factor de riesgo. (Más probable cuando es presente)

Analizamos las variables factorizadas

```

cols<-c("Gender", "Customer_Type", "Type_Travel", "Class")
ORs<-function(x, y){
  table_<-table(x,y)
  prob_1<-table_[1] / sum(table_[1:2])
  prob_2<-table_[3] / sum(table_[3:4])
  or<-prob_1 / prob_2
  factor<-colnames(table_)[1]
  return (c(factor,or))
}

```

Para cada columna llamo a la función creada

```
for (i in cols){  
  val<-ORs(data$satisfaction_re,data[,i])  
  cat(val[1], ": ", val[2], "\n")  
}
```

```
## Female : 0.622620497219871  
## disloyal Customer : 1.98180359340719  
## Business travel : 0.780324265484942  
## Business : 0.479416826019671
```

Vemos que en la variable **Gender** la probabilidad de encontrar una mujer (*Female*) satisfecha con el vuelo es mayor que la de un hombre. OR < 1. Factor de protección.

En la variable **Customer_Type** la probabilidad de encontrar un cliente no leal (*disloyal Customer*) satisfecho con el vuelo es menor que la de uno que es leal. OR > 1. Factor de Riesgo.

Vemos que en la variable **Type_Travel** la probabilidad de encontrar un viajero con opción Business (*Business travel*) satisfecho con el vuelo es mayor que la de uno que no. OR < 1. Factor de protección.

Vemos que en la variable **Class** la probabilidad de encontrar un viajero con opción Business (*Business*) satisfecho con el vuelo es mayor que la de uno que no. OR < 1. Factor de protección.

2.3 Cálculo de las OR (Odds-Ratio)

Similar al ejercicio anterior llamo a la función creada para cada variable.

Class

```
val<-ORs(data$satisfaction_re,data[, "Class"])  
cat(val[1], ": ", val[2], "\n")
```

```
## Business : 0.479416826019671
```

Vemos que en la variable **Class** la probabilidad de encontrar un viajero con opción Business (*Business*) satisfecho con el vuelo es mayor que la de uno que no. OR < 1. Factor de protección.

Customer_Type

```
val<-ORs(data$satisfaction_re,data[, "Customer_Type"])  
cat(val[1], ": ", val[2], "\n")
```

```
## disloyal Customer : 1.98180359340719
```

En la variable **Customer_Type** la probabilidad de encontrar un cliente no leal (*disloyal Customer*) satisfecho con el vuelo es menor que la de uno que es leal. OR > 1. Factor de Riesgo.

Gender

```
val<-ORs(data$satisfaction_re,data[, "Gender"])  
cat(val[1], ": ", val[2], "\n")
```

```
## Female : 0.622620497219871
```

Vemos que en la variable **Gender** la probabilidad de encontrar una mujer (*Female*) satisfecha con el vuelo es mayor que la de un hombre. OR < 1. Factor de protección.

Ent

```
table_<-table(data$satisfaction_re,data$Ent)
prob_1<-table_[7] / sum(table_[7:8]) + table_[9] / sum(table_[9:10])
prob_2<-table_[1] / sum(table_[1:2]) + table_[3] / sum(table_[3:4]) + table_[5] / sum(table_[5:6]) + ta
prob_1 / prob_2

## [1] 0.5388936
```

Vemos que en la variable **Ent** la probabilidad de encontrar un pasajero que haya valorado el entretenimiento con 3 o 4 en el vuelo es menor que la de una valoración de 0,1,2 o 5 juntos. OR < 1. Factor de protección.

2.4 Matriz de confusión

A continuación analizad la precisión de ModlgF, comparando la predicción del modelo contra el conjunto de prueba (testing). Se asumirá que la predicción del modelo es 1, satisfied, si la probabilidad del modelo de regresión logística es superior o igual a 0.5 y 0 en caso contrario. Analizad la matriz de confusión y las medidas de sensibilidad y especificidad.

```
predictions<-predict(modlgF, test[,3:20], type="response") # Response para conseguir valores entre 0 y 1
TP<-sum(test[which(predictions>=0.5),22] == 1)
FP<-sum(test[which(predictions>=0.5),22] == 0)
FN<-sum(test[which(predictions<0.5),22] == 1)
TN<-sum(test[which(predictions<0.5),22] == 0)
first_c <- c(TP, FP)
second_c <- c(FN, TN)
CM <- data.frame(first_c, second_c)
rownames(CM)<-c("TRUE", "FALSE")
colnames(CM)<-c("TRUE", "FALSE")
CM

##          TRUE FALSE
## TRUE    12071  2196
## FALSE   2170   9368
```

Vemos en la Matriz de confusión:

- TP: 12071
- FP: 2196
- FN: 2170
- TN: 9368

$$Sensibilidad = TP/(TP + FN)$$

```
sensibilidad<-TP/(TP+FN)
sensibilidad
```

```
## [1] 0.8460784
```

$$Especificidad = TN / (TN + FP)$$

```
especificidad<-TN/(TN+FP)
especificidad
```

```
## [1] 0.8119258
```

2.5 Predicción

Según ModlgF, calculad la probabilidad de que el cliente encuestado número tres (tercera fila de la base de datos) estuviera o no satisfecho con la aerolínea.

```
predict(modlgF, test[3,3:20], type="response")
```

```
##      7
## 0.87865
```

Tiene una probabilidad de 87.9%. Y efectivamente es un viajero satisfecho:

```
test[3,20:22] # Printeo 2 columnas demás para mejor visualización
```

```
##   Departure_Delay Arrival_Delay satisfaction_re
## 7           38            0             1
```

2.6 Bondad del ajuste

- a) Evaluad la bondad del ajuste, mediante la devianza. Para que ModlgF sea bueno la devianza residual debe ser menor que la devianza nula. En ese caso el modelo predice la variable dependiente con mayor precisión.
- b) Evaluad la eficacia del modelo según el test Chi-cuadrado. En este caso el valor del estadístico Chicuadrado observado es igual a la diferencia de devianzas (nula-residual). Calculad la probabilidad asociada al estadístico del contraste utilizando la función pchisq.

a)

```
summary(modlgF)
```

```
##
## Call:
## glm(formula = satisfaction_re ~ Gender + Customer_Type + Age +
##       Type_Travel + Class + Distance + Seat_comfort + Food_drink +
##       Gate + Wifi + Ent + Ease_booking + Service + Baggage_handling +
##       Checkin_service + Cleanliness + Online_boarding + Departure_Delay,
##       family = binomial, data = train)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q        Max
## -2.9699   -0.5896    0.2037   0.5409   3.5762
##
```

```

## Coefficients:
##                               Estimate Std. Error z value Pr(>|z|)
## (Intercept)           -6.286e+00 7.025e-02 -89.475 < 2e-16 ***
## GenderMale            -1.019e+00 1.822e-02 -55.947 < 2e-16 ***
## Customer_TypeLoyal   1.895e+00 2.746e-02  69.024 < 2e-16 ***
## Age                  -7.390e-03 6.300e-04 -11.730 < 2e-16 ***
## Type_TravelPersonal  -8.329e-01 2.583e-02 -32.250 < 2e-16 ***
## ClassEco              -7.622e-01 2.358e-02 -32.326 < 2e-16 ***
## ClassEco_Plus         -8.514e-01 3.615e-02 -23.549 < 2e-16 ***
## Distance             -1.247e-04 9.486e-06 -13.145 < 2e-16 ***
## Seat_comfort          2.768e-01 1.003e-02  27.615 < 2e-16 ***
## Food_drink            -2.880e-01 9.974e-03 -28.871 < 2e-16 ***
## Gate                 3.883e-02 7.983e-03  4.864 1.15e-06 ***
## Wifi                 -8.751e-02 9.641e-03 -9.077 < 2e-16 ***
## Ent                  7.180e-01 8.875e-03  80.910 < 2e-16 ***
## Ease_booking          3.354e-01 1.227e-02  27.331 < 2e-16 ***
## Service              3.188e-01 9.151e-03  34.841 < 2e-16 ***
## Baggage_handling     1.274e-01 1.030e-02  12.369 < 2e-16 ***
## Checkin_service       2.930e-01 7.644e-03  38.335 < 2e-16 ***
## Cleanliness          8.889e-02 1.070e-02  8.304 < 2e-16 ***
## Online_boarding      1.686e-01 1.057e-02  15.946 < 2e-16 ***
## Departure_Delay      -5.032e-03 2.520e-04 -19.970 < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
## Null deviance: 142793  on 103640  degrees of freedom
## Residual deviance: 81475  on 103621  degrees of freedom
## AIC: 81515
##
## Number of Fisher Scoring iterations: 5

```

Podemos observar la *Null deviance* y *Residual deviance* con sus grados de libertad.

Vemos que la desvianza residual es menor a la nula. El estadístico G que mide la diferencia entre las dos devianza:

```
G<-142793-81475
```

```
G
```

```
## [1] 61318
```

También calculamos usando la función *anova()*

```
anova(modlgF,test="Chisq")
```

```

## Analysis of Deviance Table
##
## Model: binomial, link: logit
##
## Response: satisfaction_re
##
```

```

## Terms added sequentially (first to last)
##
##
##          Df Deviance Resid. Df Resid. Dev Pr(>Chi)
## NULL             103640    142793
## Gender           1     4727.6   103639   138065 < 2.2e-16 ***
## Customer_Type    1    10075.8   103638   127989 < 2.2e-16 ***
## Age              1     155.0    103637   127834 < 2.2e-16 ***
## Type_Travel      1     4850.1   103636   122984 < 2.2e-16 ***
## Class             2     5368.1    103634   117616 < 2.2e-16 ***
## Distance          1     421.7    103633   117195 < 2.2e-16 ***
## Seat_comfort      1     6495.9   103632   110699 < 2.2e-16 ***
## Food_drink         1     1795.7   103631   108903 < 2.2e-16 ***
## Gate              1     698.1    103630   108205 < 2.2e-16 ***
## Wifi               1     2774.8   103629   105430 < 2.2e-16 ***
## Ent                1    11177.8   103628   94252 < 2.2e-16 ***
## Ease_booking        1    7005.0    103627   87247 < 2.2e-16 ***
## Service            1    2851.6    103626   84396 < 2.2e-16 ***
## Baggage_handling   1     422.6    103625   83973 < 2.2e-16 ***
## Checkin_service     1    1789.6    103624   82183 < 2.2e-16 ***
## Cleanliness         1      59.9    103623   82124 9.998e-15 ***
## Online_boarding     1     237.2    103622   81886 < 2.2e-16 ***
## Departure_Delay     1     411.8    103621   81475 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

Vemos que todas las variables son significativas por el p-value

b)

```
sum(residuals(modlgF, type="pearson")^2)
```

```
## [1] 130856.7
```

```
1-pchisq(sum(residuals(modlgF, type="pearson")^2), 1)
```

```
## [1] 0
```

Apreciamos que el modelo sigue siendo significativo ya que el resultado es menor a la suma de cuadrados de los residuos de Pearson

2.7 Curva ROC

Dibujad la curva ROC y calcular el área debajo de la curva con Modlg. Discutid el resultado

```
library(pROC)
```

```
## Type 'citation("pROC")' for a citation.
```

```
##
## Attaching package: 'pROC'
```

```

## The following objects are masked from 'package:stats':
##
##     cov, smooth, var

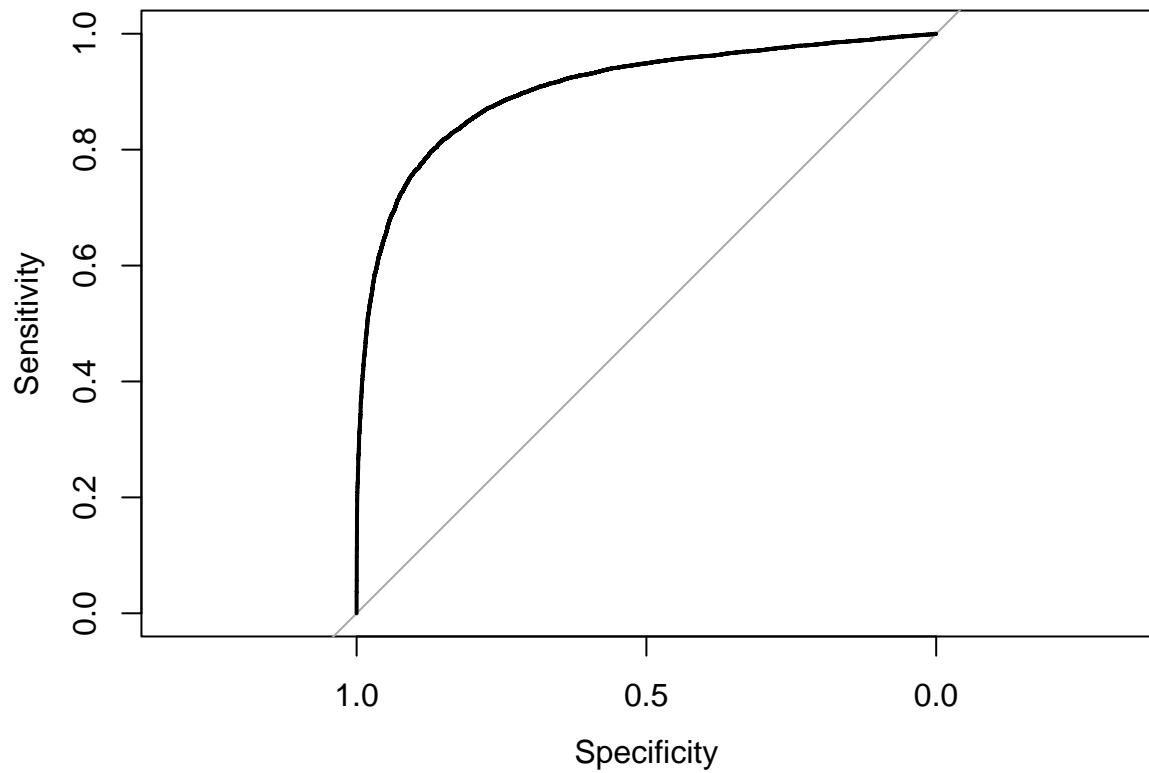
r=roc(test$satisfaction_re,predict(modlgF, test,type="response"),data=test)

## Setting levels: control = 0, case = 1

## Setting direction: controls < cases

plot(r)

```



En la curva ROC vemos la sensibilidad del modelo sobre la especificidad

```
auc(r)
```

```
## Area under the curve: 0.9048
```

El área bajo la curva es de 0.9048, por lo que el modelo creado acierta en la mayoría de ocasiones.

3 Informe Ejecutivo

Realización del informe ejecutivo

3.1 Presentación de los principales resultados del estudio en una tabla

```
p0<-c("Linear Regresión. Dependiente Arrival_delay, Independiente: Distance", "p-value de 2.2e-16","esta")
p1<-c("Añadimos variable Departure_Delay", "coeficiente de determinación ajustado de 0.9227","Modelo pr
p2<-c("Añadimos Food_drink, Service, satisfaction y Customer_Type", "coeficiente de determinación ajust
p3<-c("Comprobación colinealidad modelF", "FIV es cercano a 1 por lo que es menor a 12.95 (FIV general)
p4<-c("Predicción viajero", "2500 miles, 30 min departure_delay y satisfaction 2","retraso en la llegada
p5<-c("Estudiar colinealidad y correlación de modelo con todas las variables", "colinealidad entre Depa
p6<-c("Estudio OR de class", "valor de 0.479","OR < 1 / Factor de protección")
p7<-c("Estudio OR de Customer_Type", "valor de 1.982","OR > 1 / Factor de riesgo")
p8<-c("Estudio OR de Gender", "valor de 0.623","OR < 1 / Factor de protección")
p9<-c("Estudio OR de Ent", "valor de 0.538","OR < 1 / Factor de protección")
p10<-c("Matriz confusión modlgF", "TP: 127071, FP: 2196, FN: 2170, TN: 9368","Sensibilidad: 0.846 / Esp
p11<-c("Predicción modelgF cliente encuestado número 3", "Probabilidad de 87.9% de estar satisfecho","E
p12<-c("ROC del modelo", "El área bajo la curva es de 0.9048","Modelo preciso")
table<-data.frame(p1,p2,p3,p4,p5,p6,p7,p8,p9,p10,p11,p12)
knitr::kable(t(table))
```

p1	Añadimos variable Departure_Delay	coeficiente de determinación ajustado de 0.9227	Modelo predictivo bueno y hay mejora
p2	Añadimos Food_drink, Service, satisfaction y Customer_Type	coeficiente de determinación ajustado de 0.9228	Descartamos Food_drink (no significativa) y Customer_Type(confusa)
p3	Comprobación colinealidad modelF	FIV es cercano a 1 por lo que es menor a 12.95 (FIV general)	No encontramos problemas de colinealidad
p4	Predicción viajero	2500 miles, 30 min departure_delay y satisfaction 2	retraso en la llegada del vuelo de 29.84 minutos. (29 minutos y 50 segundos) Descartamos Arrival_Delay y satisfaction_re
p5	Estudiar colinealidad y correlación de modelo con todas las variables	colinealidad entre Departure_Delay y Arrival_Delay. Correlación 1 a 1 de satisfaction y satisfaction_re	
p6	Estudio OR de class	valor de 0.479	OR < 1 / Factor de protección
p7	Estudio OR de Customer_Type	valor de 1.982	OR > 1 / Factor de riesgo
p8	Estudio OR de Gender	valor de 0.623	OR < 1 / Factor de protección
p9	Estudio OR de Ent	valor de 0.538	OR < 1 / Factor de protección
p10	Matriz confusión modlgF	TP: 127071, FP: 2196, FN: 2170, TN: 9368	Sensibilidad: 0.846 / Especificidad: 0.812
p11	Predicción modelgF cliente encuestado número 3	Probabilidad de 87.9% de estar satisfecho	Efectivamente está satisfecho
p12	ROC del modelo	El área bajo la curva es de 0.9048	Modelo preciso

3.2 Resumen ejecutivo. Conclusiones del análisis

- Se puede concluir que para la predicción del retraso en la llegada, se puede conseguir un modelo que predice bien usando solo dos variables. Estas son retraso a la hora de salida y distancia al destino.
- Para este modelo se puede utilizar una regresión lineal. Que indica que existe una relación lineal entre las variables independientes (distancia y retraso en la salida) y la variable dependiente (retraso en la llegada).
- Este modelo nos devolverá los minutos que predice de retraso.
- Para el segundo modelo, predecimos si el cliente estará o no satisfecho. Para ello utilizamos todos los valores recogidos por los clientes en el pasado menos el tiempo de retraso en el despegue, la valoración de bebida y comida y el tipo de cliente que es. El motivo es que estas variables no son de interés en nuestro modelo.
- El modelo nos devolverá una probabilidad entre 0 y 1 de cuánto de probable es que el cliente esté satisfecho.
- La curva ROC, nos indica que este modelo es bueno a la hora de predecir.