Demo Day Módulo 2 RStudio

PERLA CONCHITA PEÑA CAMPOS

Contenido

1. Data

2. EDA

3. Pruebas estadísticas

4. Métodos de clasificación

Data set

Decribe los datos de dos hoteles del 01 julio 2015 al 31 Agosto 2017

- H1: Hotel resort

- H2: Hotel ciudad

La variable que se estudia es la cancelación de la reservación con respecto al resto.

Se hizo limpieza de la base de datos y transformación de variables (revisar pdf DemoDay_M2)

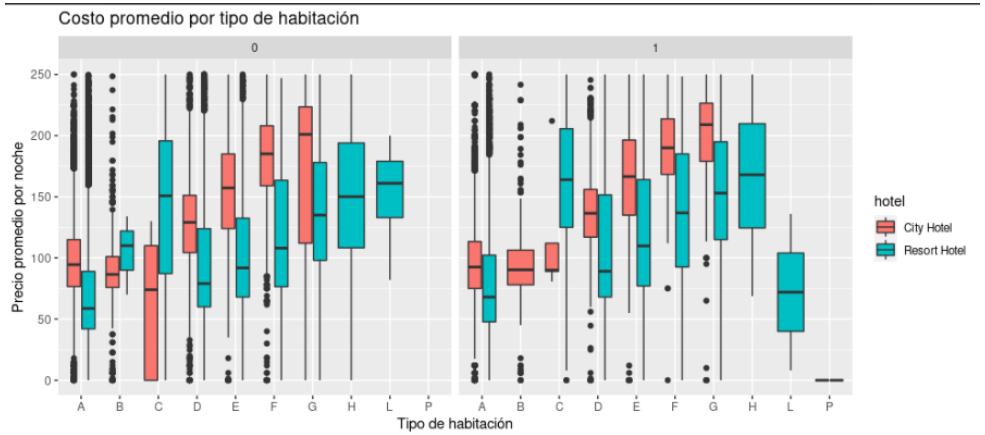
https://www.sciencedirect.com/science/article/pii/S2352340918315191

```
> str(booking)
'data.frame': 119390 obs. of  32 variables:
 $ hotel                         : chr  "Resort Hotel" "Resort Hotel" "R
 $ is_canceled                   : int  0 0 0 0 0 0 0 0 1 1 ...
 $ lead_time                     : int  342 737 7 13 14 14 0 9 85 75 ...
 $ arrival_date_year             : int  2015 2015 2015 2015 2015 2015 20
 $ arrival_date_month            : chr  "July" "July" "July" "July" ...
 $ arrival_date_week_number      : int  27 27 27 27 27 27 27 27 27 27 ..
 $ arrival_date_day_of_month     : int  1 1 1 1 1 1 1 1 1 1 ...
 $ stays_in_weekend_nights       : int  0 0 0 0 0 0 0 0 0 0 ...
 $ stays_in_week_nights          : int  0 0 1 1 2 2 2 2 3 3 ...
 $ adults                        : int  2 2 1 1 2 2 2 2 2 2 ...
 $ children                      : int  0 0 0 0 0 0 0 0 0 0 ...
 $ babies                        : int  0 0 0 0 0 0 0 0 0 0 ...
 $ meal                          : chr  "BB" "BB" "BB" "BB" ...
 $ country                       : chr  "PRT" "PRT" "GBR" "GBR" ...
 $ market_segment                : chr  "Direct" "Direct" "Direct" "Corp
 $ distribution_channel          : chr  "Direct" "Direct" "Direct" "Corp
 $ is_repeated_guest             : int  0 0 0 0 0 0 0 0 0 0 ...
 $ previous_cancellations        : int  0 0 0 0 0 0 0 0 0 0 ...
 $ previous_bookings_not_canceled: int  0 0 0 0 0 0 0 0 0 0 ...
 $ reserved_room_type            : chr  "C" "C" "A" "A" ...
 $ assigned_room_type            : chr  "C" "C" "C" "A" ...
 $ booking_changes               : int  3 4 0 0 0 0 0 0 0 0 ...
 $ deposit_type                  : chr  "No Deposit" "No Deposit" "No De
```

EDA

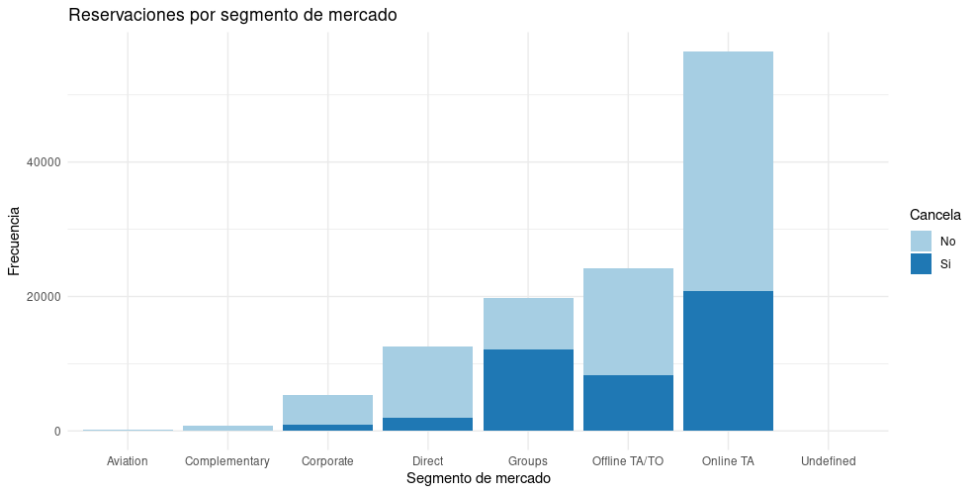
Costo promedio por habitación

```
b1 %>%
  ggplot(aes(x=reserved_room_type, y = adr, fill = hotel))+
  geom_boxplot()+
  facet_wrap("is_canceled") +
  ylim(0,250)+
  ggtitle("Costo promedio por tipo de habitación") +
  xlab("Tipo de habitación") +
  ylab("Precio promedio por noche")+
  theme(plot.title = element_text(size=12))+
  theme_gray()
```



Frecuencia (tipo de mercado)

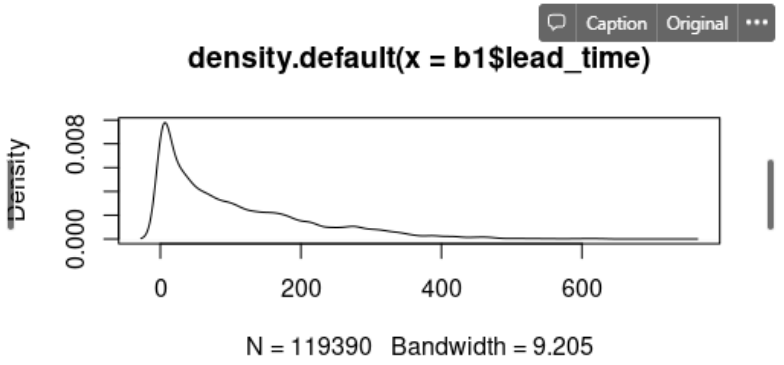
```
b1 %>%
  ggplot(aes(x=market_segment, y=frequency(market_segment), fill=is_canceled)) +
  geom_bar(stat="identity")+
  scale_fill_brewer(palette="Paired", name = "Cancela", labels = c("No", "Si"))+
  theme_minimal()+
  ggtitle("Reservaciones por segmento de mercado") +
  xlab("Segmento de mercado") +
  ylab("Frecuencia")
```



Prueba estadística de hipótesis

Prueba de normalidad *ejemplo de una variable

```r
#Verificar la normalidad de los datos con un histograma
hist(b1$lead_time)

#Densidad
d <- density(b1$lead_time)
plot(d)

#Prueba de  shapiro test
lt.test <- shapiro.test(b1$lead_time[0:5000])
lt.test

Shapiro-Wilk normality test

data:  b1$lead_time[0:5000]
W = 0.88581, p-value < 2.2e-16
```



Clasificación

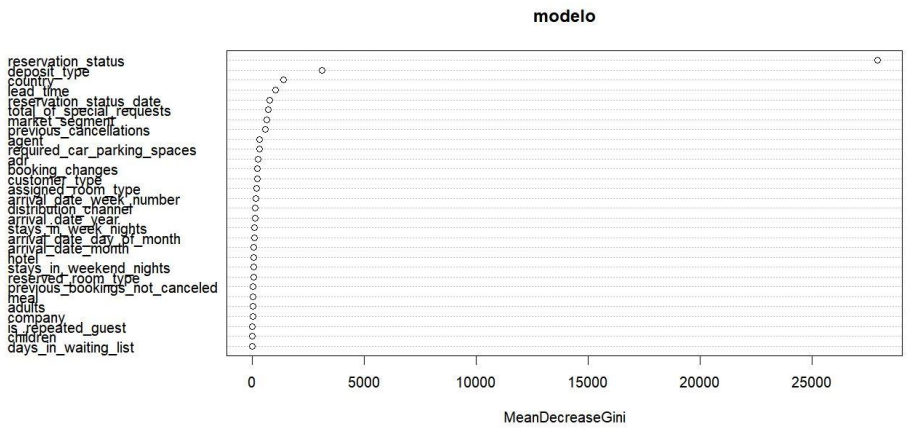
RANDOM FOREST, REGRESIÓN LOGÍSTICA Y SVM

Random Forest

```
#Crear semilla, datos test y datos tain
set.seed(101)
tamano.total <- nrow(bforest)
tamano.entreno <- round(tamano.total*0.7)
datos.indices <- sample(1:tamano.total , size=tamano.entreno)
datos.entreno <- bforest[datos.indices,]
datos.test <- bforest[-datos.indices,]

#Modelo Random Forest
modelo <- randomForest(is_canceled~., data=datos.test)
modelo

#gráficos y resultados
varImpPlot(modelo)
plot(modelo)
legend("right", colnames(modelo$err.rate), lty = 1:5, col = 1:6)
importance(modelo2)

# Separar árboles
> split_var_1 <- sapply(seq_len(modelo$ntree),
+                       function(i) getTree(modelo, i, labelVar=TRUE)[1, "split var"])
> table(split_var_1)
split_var_1
```

```
Call:
 randomForest(formula = is_canceled ~ ., data = datos.test
               Type of random forest: classification
                     Number of trees: 500
No. of variables tried at each split: 5

        OOB estimate of  error rate: 0.01%
Confusion matrix:
      0     1  class.error
0 22522     0 0.0000000000
1     2 13293 0.0001504325
> |
```

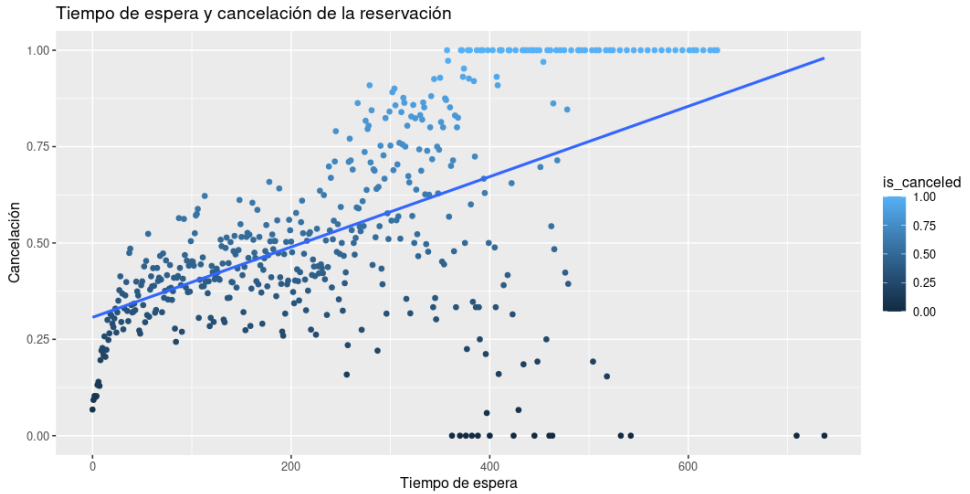
varImPlot



- Se rescatan del modelo:
  - deposit_type
  - country
  - lead_time
  - total_of_special_requests

EDA random forest
*ejemplo: tiempo de espera

```r
#agrupar por lead_time (Tiempo de espera)
grouplead_time <- aggregate(booking["is_canceled"], by=booking["lead_time"], mean)
grouplead_time

grouplead_time %>%
  ggplot(aes(x=lead_time, y = (is_canceled), color = is_canceled))+
  geom_point()+
  geom_smooth(method = "lm", se = FALSE)+
  ggtitle("Tiempo de espera y cancelación de la reservación") +
  xlab("Tiempo de espera") +
  ylab("Cancelación")

#agrupar por deposit_type (Tipo de deposito)
groupdeposit_type <- aggregate(booking["is_canceled"], by=booking["deposit_type"], mean)
groupdeposit_type
```



Regresión logística

```r
#set.seed(1)
df <- bforest
nobs <- nrow(bforest)
itrain <- sample(nobs, 0.8 * nobs)
train <- df[itrain, ]
test <- df[-itrain, ]

#Regresión logística con todas las variables
rl <- glm(is_canceled ~., data = train)
rl
summary(rl)
#Regresión logística con las variables de random forest
rl2 <- glm(is_canceled ~ deposit_type + country + lead_time +
           market_segment + adr, family = binomial, data = train)
rl2
summary(rl2)
#Ajuste de la regresión logística sin la variable que no aporta
rl3 <- update(rl2, ~. -country)
summary(rl3)
#Ajuste de la regresión logística con la variable faltante
rl4 <- update(rl3, ~. +total_of_special_requests)
summary(rl4)
#coeficientes
rl4$coef
plot(rl4)
```

```
                                Estimate Std. Error z value
(Intercept)                   -1.540e+00  1.724e-01  -8.932
deposit_typeNon Refund         6.123e+00  1.186e-01  51.631
deposit_typeRefundable        -2.658e-01  2.067e-01  -1.286
lead_time                      3.925e-03  8.776e-05  44.724
market_segmentComplementary   -4.413e-01  2.117e-01  -2.084
market_segmentCorporate       -6.177e-01  1.779e-01  -3.472
market_segmentDirect          -7.840e-01  1.740e-01  -4.507
market_segmentGroups          -2.633e-01  1.739e-01  -1.514
market_segmentOffline TA/TO   -7.958e-01  1.734e-01  -4.591
market_segmentOnline TA        3.001e-01  1.720e-01   1.745
market_segmentUndefined        1.106e+01  5.124e+01   0.216
adr                            2.958e-03  1.710e-04  17.295
                              Pr(>|z|)
(Intercept)                    < 2e-16 ***
deposit_typeNon Refund         < 2e-16 ***
deposit_typeRefundable        0.198390
lead_time                      < 2e-16 ***
market_segmentComplementary   0.037156 *
market_segmentCorporate       0.000517 ***
market_segmentDirect          6.59e-06 ***
market_segmentGroups          0.130061
market_segmentOffline TA/TO   4.41e-06 ***
market_segmentOnline TA       0.081027 .
market_segmentUndefined       0.829173
adr                            < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Deposit_type, lead_time, market_segment, adr

SVM

```r
# Partición de los datos
set.seed(1)
df <- bforest
nobs <- nrow(bforest)
itrain <- sample(nobs, 0.8 * nobs)
train <- df[itrain, ]
test <- df[-itrain, ]

#Instalar paquete
install.packages("kernlab")
library(kernlab)
library(e1071)


## Preparar datos
set.seed(2022)
train = sample(nrow(bforest),
               round(nrow(bforest)/2))
tail(bforest[train, ])

# Modelo con todo el data set
best <- svm(is_canceled~.,  data = bforest[train,],
            kernel = "radial",
            cost = 100,
            gamma = 1.51)
```

```r
# svm con lead time y adr
best2 <- svm(is_canceled ~ lead_time + adr,  data = train,
             kernel = "radial",
             cost = 100,
             gamma = 1.51)
best2
summary(best2)
# resultados
names(best2)
# plot del modelo de svm
plot(best2, test, lead_time ~ adr)

plot(best2, train, lead_time ~ adr)

#Matriz de confusión con modelo actual y predictivo
table(train$is_canceled, fitted(best2), dnn = c("Actual", "Predicho"))
pred <- predict(best2, test)
pred
table(test$is_canceled, pred)

#tuning
set.seed(2)
tune.out <- tune(svm, is_canceled ~ lead_time + adr, data = train,
     kernel = "radial",
     ranges = list(
       cost = c(0.1, 1, 10, 100, 1000),
       gama = c(0.5, 1, 2, 3, 4)
     ))
```

Matriz de confusión y plot SVM

```
Call:
svm(formula = is_canceled ~ lead_time + adr, data = train,
    kernel = "radial", cost = 100, gamma = 1.51)


Parameters:
   SVM-Type:  C-classification
 SVM-Kernel:  radial
       cost:  100

Number of Support Vectors:  63100

 ( 31901 31199 )


Number of Classes:  2

Levels:
 0 1
```

```
table(train$is_canceled, fitted(best2), dnn = c("Actual", "Predicho"))
      Predicho
Actual     0     1
     0 55424  4755
     1 25772  9561

table(test$is_canceled, pred)
predict
pred
       0     1
  0 13854  1133
  1  6597  2294
```