Pronostico Demanda

#### Pilar Prado

#### 24/3/2021

UNIVERSIDAD CATOLICA BOLIVIANA “SAN PABLO”



Caption for the picture.

GRUPO DE TRABAJO:

Pilar Prado

Daniela Valdivia

.

.

PROBLEMA: SE REQUIERE HACER UN PRONOSTICO DE DEMANDA DE UNA EMPRESA DE DELIVERY

El cliente es una empresa de entrega de comidas que opera en varias ciudades. Tienen varios centros logísticos en estas ciudades para enviar pedidos de comida a sus clientes. El cliente quiere un pronóstico de la demanda para las próximas semanas para que estos centros planifiquen el stock de materias primas.

La reposición de la mayoría de las materias primas se realiza semanalmente y, dado que la materia prima es perecedera, la planificación de las adquisiciones es de suma importancia.

DATA SET

library(readr)  
library(ggplot2)  
  
train <- read\_csv("D:/TRABAJOR/train.csv")

##   
## -- Column specification --------------------------------------------------------  
## cols(  
## id = col\_double(),  
## week = col\_double(),  
## center\_id = col\_double(),  
## meal\_id = col\_double(),  
## checkout\_price = col\_double(),  
## base\_price = col\_double(),  
## emailer\_for\_promotion = col\_double(),  
## homepage\_featured = col\_double(),  
## num\_orders = col\_double()  
## )

test <- read\_csv("D:/TRABAJOR/test.csv")

##   
## -- Column specification --------------------------------------------------------  
## cols(  
## id = col\_double(),  
## week = col\_double(),  
## center\_id = col\_double(),  
## meal\_id = col\_double(),  
## checkout\_price = col\_double(),  
## base\_price = col\_double(),  
## emailer\_for\_promotion = col\_double(),  
## homepage\_featured = col\_double()  
## )

Leer datos filas y columnas del archivo train que contiene los datos de una semana

nrow(train)

## [1] 456548

ncol(train)

## [1] 9

names(train)

## [1] "id" "week" "center\_id"   
## [4] "meal\_id" "checkout\_price" "base\_price"   
## [7] "emailer\_for\_promotion" "homepage\_featured" "num\_orders"

head(train, 20)

## # A tibble: 20 x 9  
## id week center\_id meal\_id checkout\_price base\_price emailer\_for\_promot~  
## <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 1379560 1 55 1885 137. 152. 0  
## 2 1466964 1 55 1993 137. 136. 0  
## 3 1346989 1 55 2539 135. 136. 0  
## 4 1338232 1 55 2139 340. 438. 0  
## 5 1448490 1 55 2631 244. 242. 0  
## 6 1270037 1 55 1248 251. 252. 0  
## 7 1191377 1 55 1778 183. 184. 0  
## 8 1499955 1 55 1062 182. 183. 0  
## 9 1025244 1 55 2707 193. 192. 0  
## 10 1054194 1 55 1207 326. 384. 0  
## 11 1469367 1 55 1230 323. 390 0  
## 12 1029333 1 55 2322 322. 388 0  
## 13 1446016 1 55 2290 311. 310. 0  
## 14 1244647 1 55 1727 445. 446. 0  
## 15 1378227 1 55 1109 265. 298. 1  
## 16 1181556 1 55 2640 282. 281. 0  
## 17 1313873 1 55 2306 244. 341. 0  
## 18 1067069 1 55 2126 486 485 0  
## 19 1058482 1 55 2826 307. 306. 0  
## 20 1240935 1 55 1754 289. 289. 0  
## # ... with 2 more variables: homepage\_featured <dbl>, num\_orders <dbl>

tail(train, 20)

## # A tibble: 20 x 9  
## id week center\_id meal\_id checkout\_price base\_price emailer\_for\_promot~  
## <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 1032849 145 61 1216 414. 413. 0  
## 2 1469586 145 61 2126 437. 508. 0  
## 3 1038490 145 61 2826 328. 327. 0  
## 4 1394189 145 61 1754 323. 322. 0  
## 5 1357555 145 61 1971 330. 330. 0  
## 6 1176475 145 61 1902 446. 445. 0  
## 7 1133784 145 61 1247 446. 447. 0  
## 8 1379712 145 61 1558 548. 548. 0  
## 9 1306192 145 61 2581 547. 549. 0  
## 10 1371867 145 61 1962 548. 547. 0  
## 11 1055597 145 61 1445 729. 727. 0  
## 12 1372838 145 61 2760 193. 261. 0  
## 13 1035758 145 61 1525 319. 321. 0  
## 14 1010438 145 61 2704 321. 320. 0  
## 15 1116711 145 61 2492 456. 456. 0  
## 16 1271326 145 61 1543 484. 484. 0  
## 17 1062036 145 61 2304 482. 482. 0  
## 18 1110849 145 61 2664 238. 321. 0  
## 19 1147725 145 61 2569 244. 313. 0  
## 20 1361984 145 61 2490 292. 290. 0  
## # ... with 2 more variables: homepage\_featured <dbl>, num\_orders <dbl>

summary(train)

## id week center\_id meal\_id   
## Min. :1000000 Min. : 1.00 Min. : 10.00 Min. :1062   
## 1st Qu.:1124999 1st Qu.: 39.00 1st Qu.: 43.00 1st Qu.:1558   
## Median :1250184 Median : 76.00 Median : 76.00 Median :1993   
## Mean :1250096 Mean : 74.77 Mean : 82.11 Mean :2024   
## 3rd Qu.:1375140 3rd Qu.:111.00 3rd Qu.:110.00 3rd Qu.:2539   
## Max. :1499999 Max. :145.00 Max. :186.00 Max. :2956   
## checkout\_price base\_price emailer\_for\_promotion homepage\_featured  
## Min. : 2.97 Min. : 55.35 Min. :0.00000 Min. :0.0000   
## 1st Qu.:228.95 1st Qu.:243.50 1st Qu.:0.00000 1st Qu.:0.0000   
## Median :296.82 Median :310.46 Median :0.00000 Median :0.0000   
## Mean :332.24 Mean :354.16 Mean :0.08115 Mean :0.1092   
## 3rd Qu.:445.23 3rd Qu.:458.87 3rd Qu.:0.00000 3rd Qu.:0.0000   
## Max. :866.27 Max. :866.27 Max. :1.00000 Max. :1.0000   
## num\_orders   
## Min. : 13.0   
## 1st Qu.: 54.0   
## Median : 136.0   
## Mean : 261.9   
## 3rd Qu.: 324.0   
## Max. :24299.0

str(train)

## spec\_tbl\_df [456,548 x 9] (S3: spec\_tbl\_df/tbl\_df/tbl/data.frame)  
## $ id : num [1:456548] 1379560 1466964 1346989 1338232 1448490 ...  
## $ week : num [1:456548] 1 1 1 1 1 1 1 1 1 1 ...  
## $ center\_id : num [1:456548] 55 55 55 55 55 55 55 55 55 55 ...  
## $ meal\_id : num [1:456548] 1885 1993 2539 2139 2631 ...  
## $ checkout\_price : num [1:456548] 137 137 135 340 244 ...  
## $ base\_price : num [1:456548] 152 136 136 438 242 ...  
## $ emailer\_for\_promotion: num [1:456548] 0 0 0 0 0 0 0 0 0 0 ...  
## $ homepage\_featured : num [1:456548] 0 0 0 0 0 0 0 0 0 1 ...  
## $ num\_orders : num [1:456548] 177 270 189 54 40 28 190 391 472 676 ...  
## - attr(\*, "spec")=  
## .. cols(  
## .. id = col\_double(),  
## .. week = col\_double(),  
## .. center\_id = col\_double(),  
## .. meal\_id = col\_double(),  
## .. checkout\_price = col\_double(),  
## .. base\_price = col\_double(),  
## .. emailer\_for\_promotion = col\_double(),  
## .. homepage\_featured = col\_double(),  
## .. num\_orders = col\_double()  
## .. )

.

Vamos a retirar de la tabla todas las observaciones en las que no hay informacion para la variable meal\_id

sum(is.na(train$meal\_id))

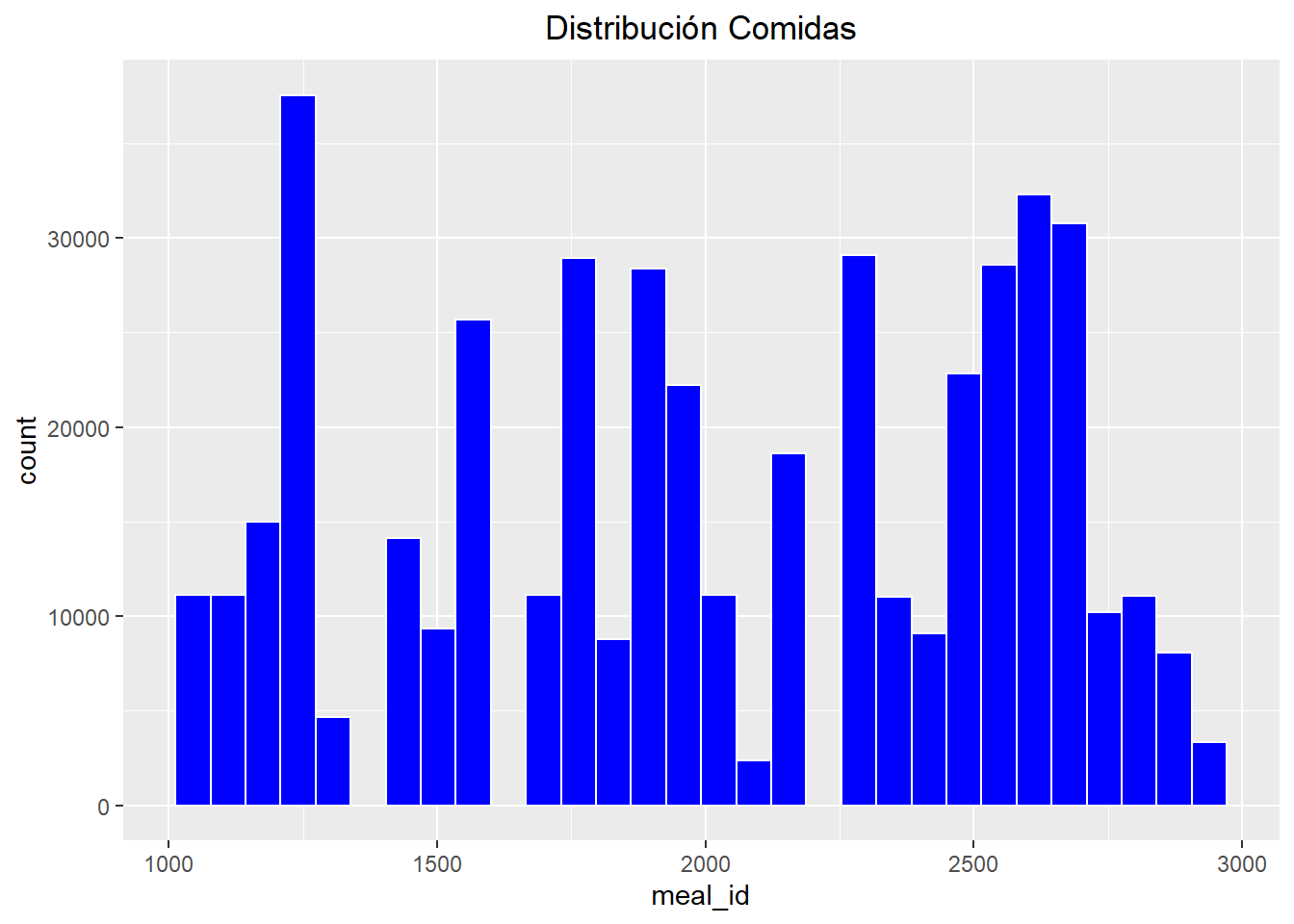
## [1] 0

.

Distribucion de la variable de respuesta

ggplot(data = train, aes(x = meal\_id)) +  
 geom\_histogram(color = "white", fill = "blue") +  
 labs(title = "Distribución Comidas") +  
 theme(plot.title = element\_text(hjust = 0.5))

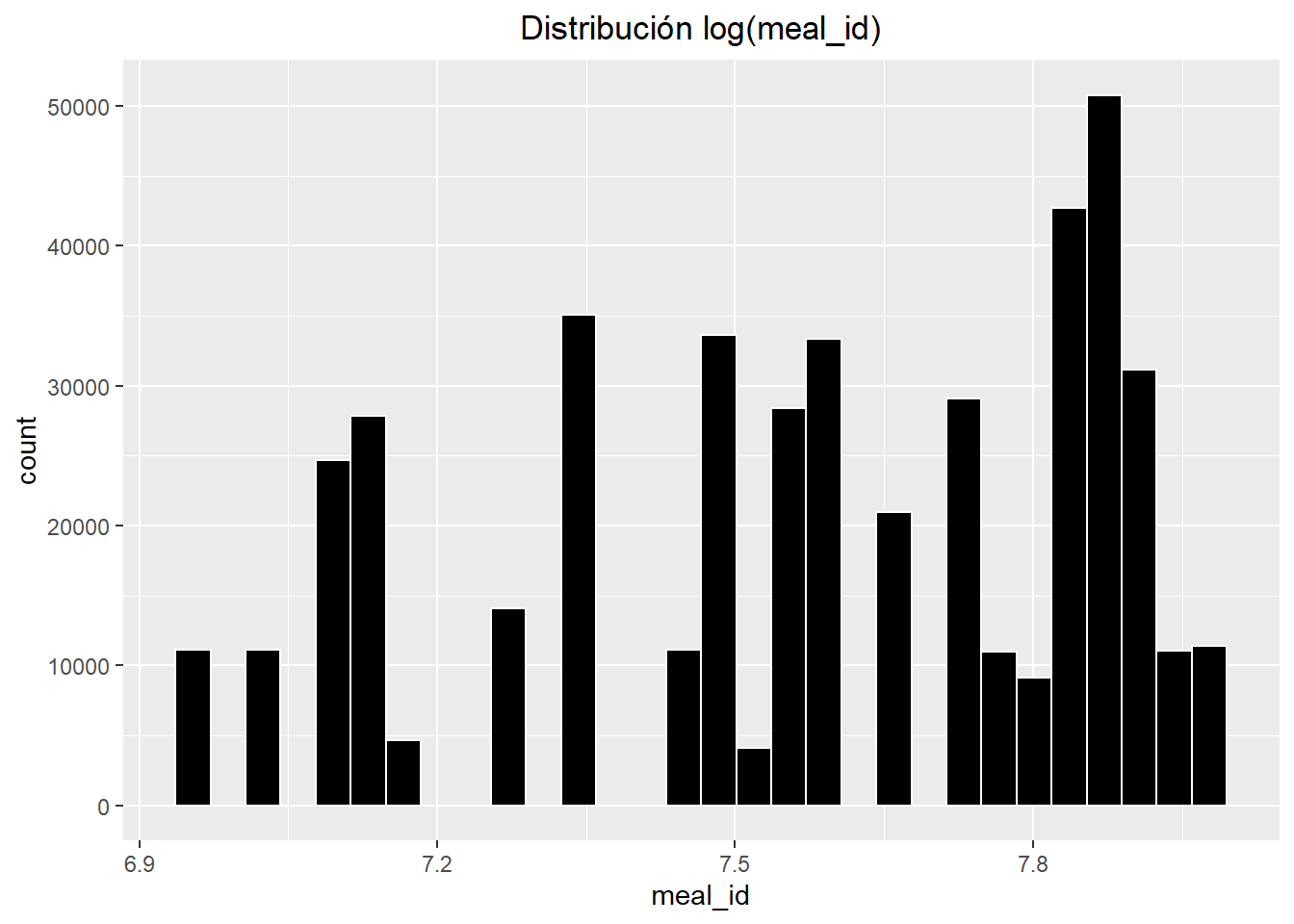
## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.



La distribucion de la variable Meal Id no es una distribución normal por lo que se procederá a normalizarla.

train$meal\_id <- log(train$meal\_id)  
  
ggplot(data = train, aes(x = meal\_id)) + geom\_histogram(color = "white", fill = "black") + labs(title = "Distribución log(meal\_id)") + theme(plot.title = element\_text(hjust = 0.5))

## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.



ARBOL DE REGRESION SIMPLE

library(tree)

## Registered S3 method overwritten by 'tree':  
## method from  
## print.tree cli

# Selección de parámetros para el árbol  
  
setup <- tree.control(nobs = nrow(train), mincut = 10, minsize = 20,mindev = 0.01)  
  
modelo\_arbolR <- tree(meal\_id ~ ., data = train, split = "deviance", control = setup)  
  
summary(modelo\_arbolR)

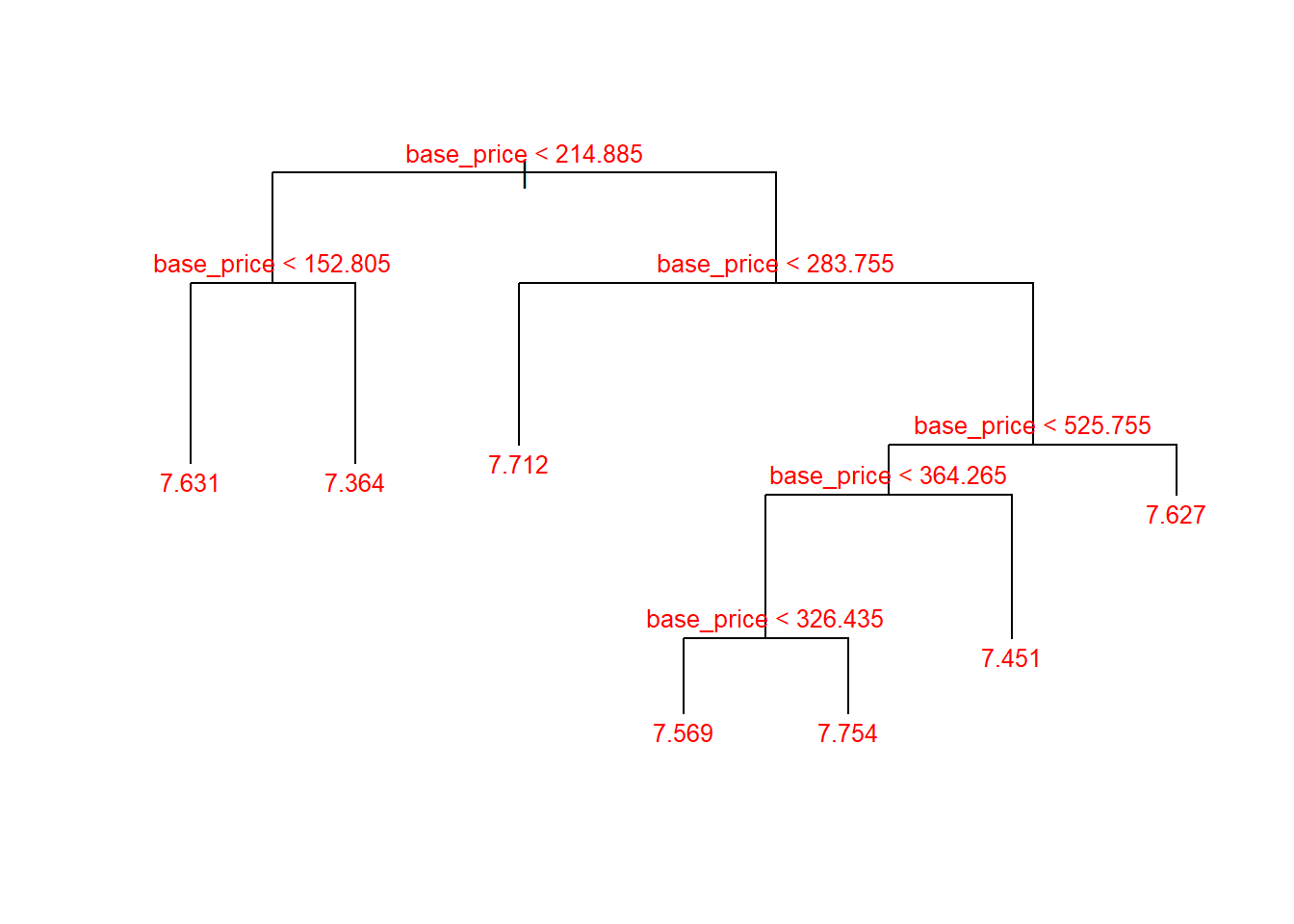
##   
## Regression tree:  
## tree(formula = meal\_id ~ ., data = train, control = setup, split = "deviance")  
## Variables actually used in tree construction:  
## [1] "base\_price"  
## Number of terminal nodes: 7   
## Residual mean deviance: 0.07224 = 32980 / 456500   
## Distribution of residuals:  
## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## -0.74300 -0.23420 0.02786 0.00000 0.20870 0.55940

.

Según la información devuelta, del conjunto de las 8 variables, el modelo ha utilizado para generar el árbol solo 1 nodos interno: base\_price. Con esta variable el número de nodos terminales es 7.

PLOTEO DEL ARBOL

plot(modelo\_arbolR, type = "proportional")   
  
  
text(modelo\_arbolR, splits = TRUE, pretty = 0, cex = 0.8, col = "red")



modelo\_arbolR

## node), split, n, deviance, yval  
## \* denotes terminal node  
##   
## 1) root 456548 39210 7.572   
## 2) base\_price < 214.885 88643 8313 7.479   
## 4) base\_price < 152.805 38455 1565 7.631 \*  
## 5) base\_price > 152.805 50188 5193 7.364 \*  
## 3) base\_price > 214.885 367905 29940 7.595   
## 6) base\_price < 283.755 79597 4940 7.712 \*  
## 7) base\_price > 283.755 288308 23610 7.562   
## 14) base\_price < 525.755 211685 18530 7.539   
## 28) base\_price < 364.265 120812 11110 7.605   
## 56) base\_price < 326.435 97218 9270 7.569 \*  
## 57) base\_price > 326.435 23594 1191 7.754 \*  
## 29) base\_price > 364.265 90873 6177 7.451 \*  
## 15) base\_price > 525.755 76623 4646 7.627 \*

PREDICCIONES DEL MODELO

predicciones <- predict(modelo\_arbolR, newdata = train)  
  
plot(x= predicciones, y = train$meal\_id, main = "Prediccion vs Real", xlab = "Predicciones", ylab = "Tipo de comida")

