Trabajo Final Data Science - Grupo 3

2023-03-15

Integrantes: José Arturo Espaillat, Johnsiel Castaños, José Delgado, Salama Mohamed-fadel Sidna.

El dataset elegido es llamado **HR Analytics: Employer Promotion Data** (https://www.kaggle.com/datasets/arashnic/hr-ana?select=test.csv) sacado de Kaggle, el cual cuenta con 13 columnas y 54808 filas. Estos datos parten de una empresa, la cual tiene un problema debido a que las promociones definitivas no se anuncian hasta después de evaluar empleado por empleo, haciendo de esto un proceso lento y tedioso. Gracias al análisis de estos históricos podemos entender cuáles son las variables que más afectan a la promoción para aumentar la eficacia del proceso, ya que ahorran mucho más tiempo al tener claro cuáles son los potenciales candidatos. También los candidatos obtienen un punto de vista de que es lo que más repercute en su promoción, pudiendo así mejorar cierto aspecto de cara a la evaluación.

De esta manera conseguimos derribar el muro de las emociones, evitando así promociones no merecidas a empleados por el mero hecho de caer bien, consiguiendo así una criticidad para que todos los empleados entiendan por qué son o no promocionados. Esto es un problema a día de hoy y vemos muchas publicaciones de como evitar conflictos entre compañeros https://www.ieie.eu/como-ascender-a-un-empleado-sin-generar-conflictos/

Personalmente, hemos querido abordar este problema porque esto es un problema real de nuestro día a día y nos pareció bastante interesante la idea de poder analizar este caso. Es cierto que esto es muy relativo a cada tipo de empresa, la cual se fija más en unas variables que en otras, pero nos puede aportar el procedimiento para extrapolarlo a otros campos. Lo que queremos lograr es plantear el análisis de problemas de este estilo, qué visualizacion nos puede ayudar, correlaciones, saber si nos podemos ahorrar pasos a través de sacar conclusiones tempranas...

En resumen, creemos que es bastante interesante el estudio de este dataset, ya que tiene cierta importancia a la hora de poder entender por qué las personas son ascendidas en su trabajo y desde el punto de vista de la empresa les beneficia en el tiempo ahorrado en estos procesos de promoción.

Para organizarnos, hemos decidido crear un repositorio en GitHub (https://github.com/Josdelser/DataScienceEmployeePromotion/tree/develop) para trabajar simultáneamente. También hemos creado una bolsa de preguntas y las hemos asignado según las capacidades y aspiraciones de cada persona. En el caso de que algún miembro quiera obtener una calificación más alta, deberá realizar un mayor número de preguntas individuales. Todos los miembros realizarán una pregunta para la parte grupal y se ha intentado que estas preguntas estén relacionadas con las preguntas individuales, para poder afrontar mejor el desafío.

Preguntas:

1. ¿Nos puede ayudar la visualización a sacar conclusiones temprana?
2. ¿Podemos averiguar qué variables afectan más a través de la predicción?
3. ¿Afectan los valores atípicos el resultado del modelo predictivo?

Cargar librerias:

```
#Packages

# install.packages("dplyr")
# install.packages("rpart")
# install.packages("rpart.plot")
# install.packages("rattle")
# install.packages("plyr")
# install.packages("ggplot2")

# libraries
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

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

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

```
library(rpart)
library(rpart.plot)
library(rattle)
```

```
## Loading required package: tibble
```

```
## Loading required package: bitops
```

```
## Rattle: A free graphical interface for data science with R.
## Version 5.5.1 Copyright (c) 2006-2021 Togaware Pty Ltd.
## Type 'rattle()' to shake, rattle, and roll your data.
```

```
library(ggplot2)
library(caret)
```

```
## Loading required package: lattice
```

```
#seed
set.seed(28)
```

Información general del Dataset:

```
data<- read.csv("train.csv")
colnames(data)
```

```
##  [1] "employee_id"          "department"           "region"
##  [4] "education"            "gender"               "recruitment_channel"
##  [7] "no_of_trainings"      "age"                  "previous_year_rating"
## [10] "length_of_service"    "awards_won."          "avg_training_score"
## [13] "is_promoted"
```

```
head(data)
```

```
##   employee_id        department    region        education gender
## 1       65438 Sales & Marketing  region_7 Master's & above      f
## 2       65141        Operations region_22       Bachelor's      m
## 3        7513 Sales & Marketing region_19       Bachelor's      m
## 4        2542 Sales & Marketing region_23       Bachelor's      m
## 5       48945        Technology region_26       Bachelor's      m
## 6       58896         Analytics  region_2       Bachelor's      m
##   recruitment_channel no_of_trainings age previous_year_rating
## 1           sourcing               1  35                    5
## 2              other               1  30                    5
## 3           sourcing               1  34                    3
## 4              other               2  39                    1
## 5              other               1  45                    3
## 6           sourcing               2  31                    3
##   length_of_service awards_won. avg_training_score is_promoted
## 1                 8           0                 49           0
## 2                 4           0                 60           0
## 3                 7           0                 50           0
## 4                10           0                 50           0
## 5                 2           0                 73           0
## 6                 7           0                 85           0
```

```
attach(data)
```

```
str(data)
```

```
## 'data.frame':    54808 obs. of  13 variables:
##  $ employee_id         : int  65438 65141 7513 2542 48945 58896 20379 16290 73202 28911 ...
##  $ department          : chr  "Sales & Marketing" "Operations" "Sales & Marketing" "Sales & Marketin
##  $ region              : chr  "region_7" "region_22" "region_19" "region_23" ...
##  $ education           : chr  "Master's & above" "Bachelor's" "Bachelor's" "Bachelor's" ...
##  $ gender              : chr  "f" "m" "m" "m" ...
##  $ recruitment_channel : chr  "sourcing" "other" "sourcing" "other" ...
##  $ no_of_trainings     : int  1 1 1 2 1 2 1 1 1 1 ...
##  $ age                 : int  35 30 34 39 45 31 31 33 28 32 ...
##  $ previous_year_rating: num  5 5 3 1 3 3 3 3 4 5 ...
##  $ length_of_service   : int  8 4 7 10 2 7 5 6 5 5 ...
##  $ awards_won.         : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ avg_training_score  : int  49 60 50 50 73 85 59 63 83 54 ...
##  $ is_promoted         : int  0 0 0 0 0 0 0 0 0 0 ...
```

```
dim(data)
```

```
## [1] 54808    13
```

```
object.size(data)/1024 #KB
```

```
## 4073.3 bytes
```

```
object.size(data)/1024^2 #MB
```

```
## 4 bytes
```

```
object.size(data)/1024^3 #GB
```

```
## 0 bytes
```

Vemos que el número de bytes del dataset es 4073,3 KB o 4,1 MB una vez lo tenemos cargado.

Es verdad que el dataset está presentable pero aún así tenemos que hacerle una etapa de transformaciones y limpieza. Primero vamos a quitar todos los NA y duplicados.

```
print(paste("Número de valores faltantes totales:", sum(is.na(data))))
```

```
## [1] "Número de valores faltantes totales: 4124"
```

```
data <- na.omit(data)
data <- unique(data)
```

Luego convertimos las variables categóricas en factores y definimos explícitamente los niveles de cada categoría.

```
dep_levels <- unique(data$department)
gen_levels <- unique(data$gender)
recru_levels <- unique(data$recruitment_channel)
promo_levels <- unique(data$is_promoted )
award_levels <- unique(data$awards_won)

data$department <- factor(data$department, levels = dep_levels)
data$gender  <- factor(data$gender , levels = gen_levels)
data$recruitment_channel <- factor(data$recruitment_channel, levels = recru_levels)
data$is_promoted <- factor(data$is_promoted, levels = promo_levels)
data$awards_won <- factor(data$awards_won, levels = award_levels)
```

Pregunta 1:

¿Nos puede ayudar la visualización a sacar conclusiones temprana?

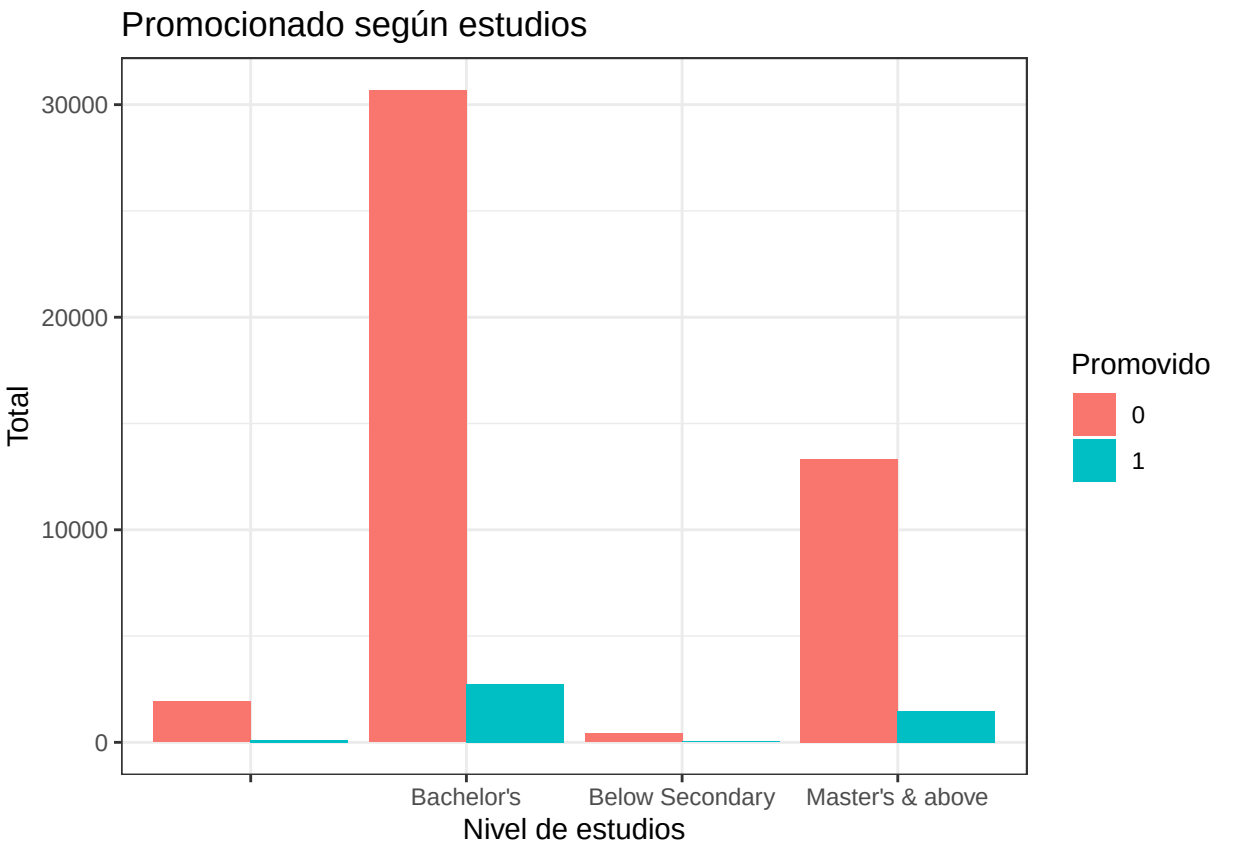
Para responder a esta pregunta vamos a realizar gráficas tanto para variables numéricas como categóricas.

Partimos de un dataset exclusivo para que no afecte a otros cálculos:

```
data_p1 <- data
```

Primero vamos a ver las variables categóricas. Empezamos por ver si los estudios influyen, vamos a graficar la distribución de empleados promovidos según sus estudios

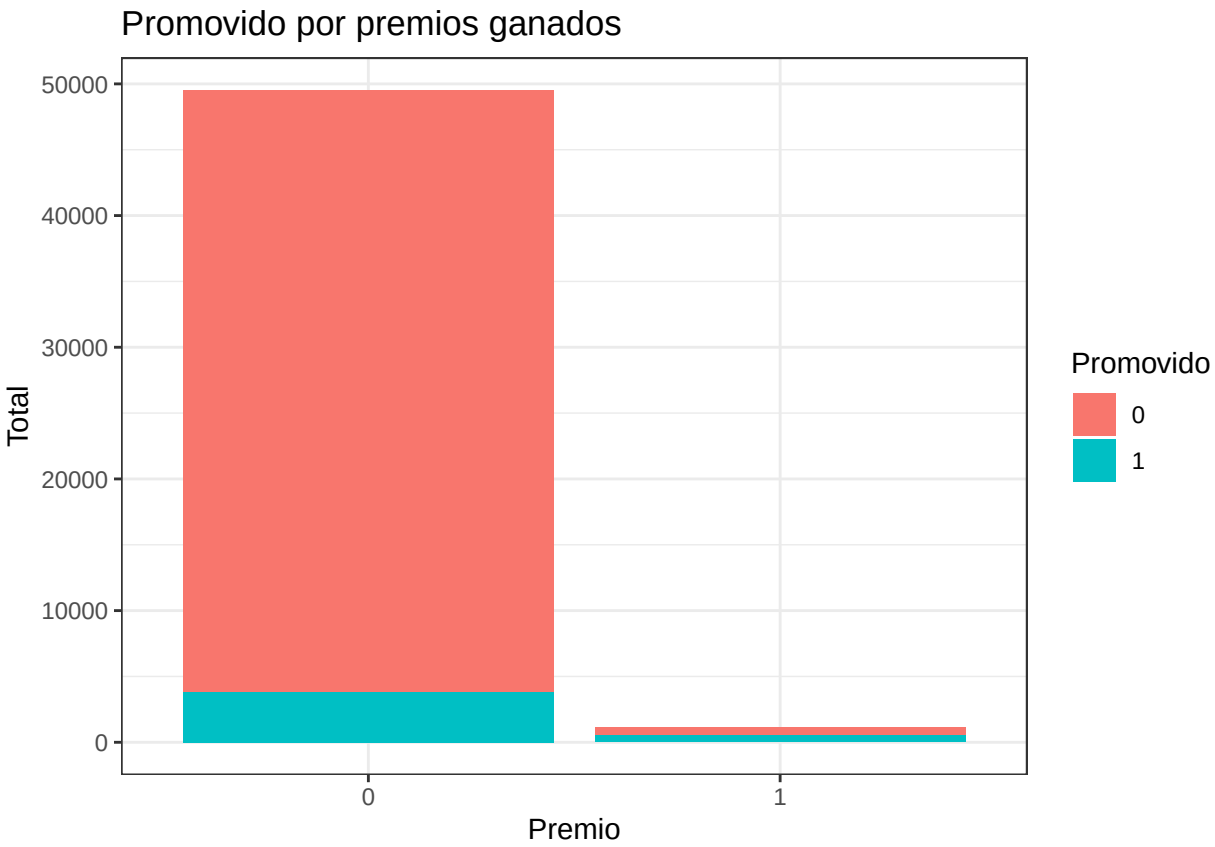
```
ggplot(data = data_p1, aes(x = education, fill = is_promoted)) +
  geom_bar(position = "dodge") +
  labs(title = "Promocionado según estudios",
       x = "Nivel de estudios",
       y = "Total",
       fill = "Promovido") +
  theme_bw()
```



Observamos que la mayoría tienen como mínimo el bachillerato y luego el Master. Podemos pensar que es una variable que afecta debido a las diferencias entre promovido o no, pero no es decisiva o tiene una importancia mayor.

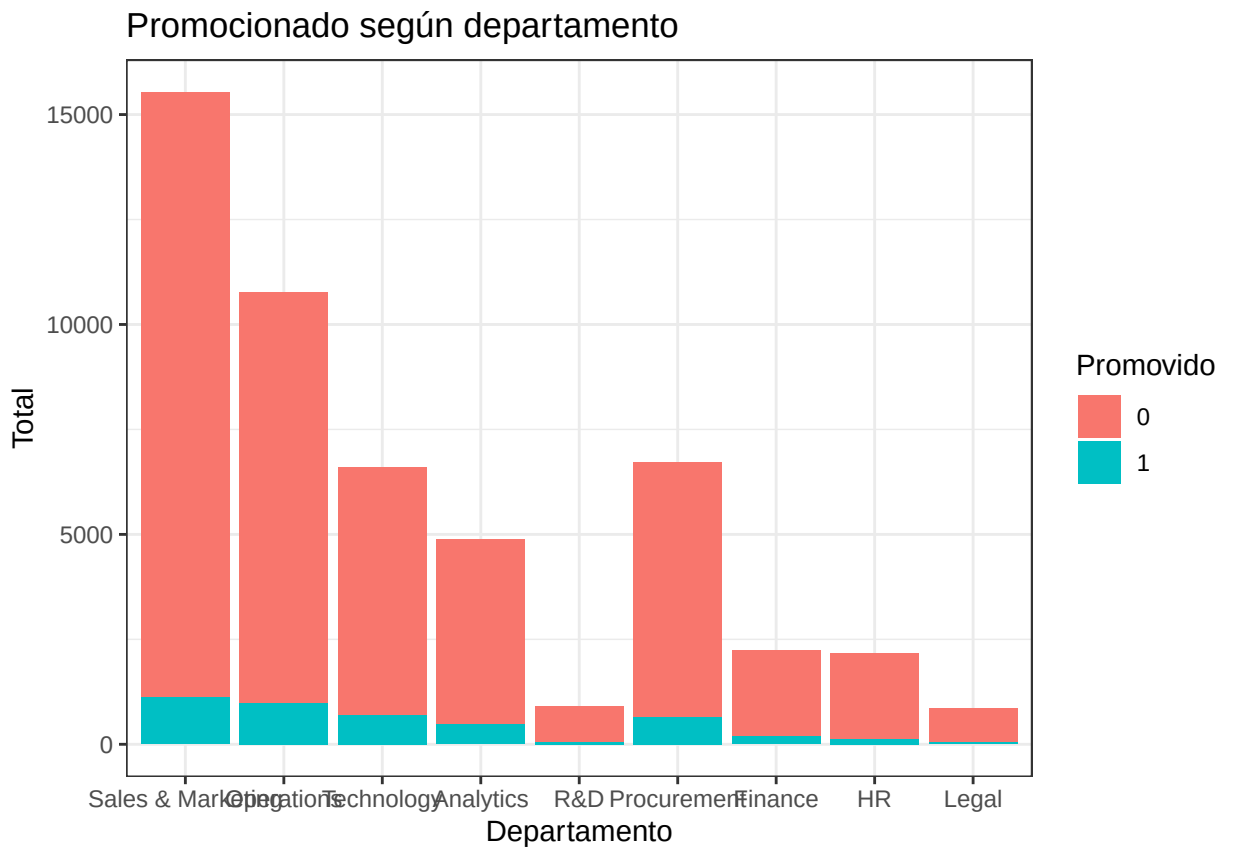
Repetimos pero con los premios ganados

```
ggplot(data = data_p1, aes(x = awards_won, fill = is_promoted)) +
  geom_bar(position = "stack") +
  labs(title = "Promovido por premios ganados",
       x = "Premio",
       y = "Total",
       fill = "Promovido") +
  theme_bw()
```



Se observa que para ser promovido es relevante tener un premio, ya que el número de promovidos con premios es notablemente mayor.

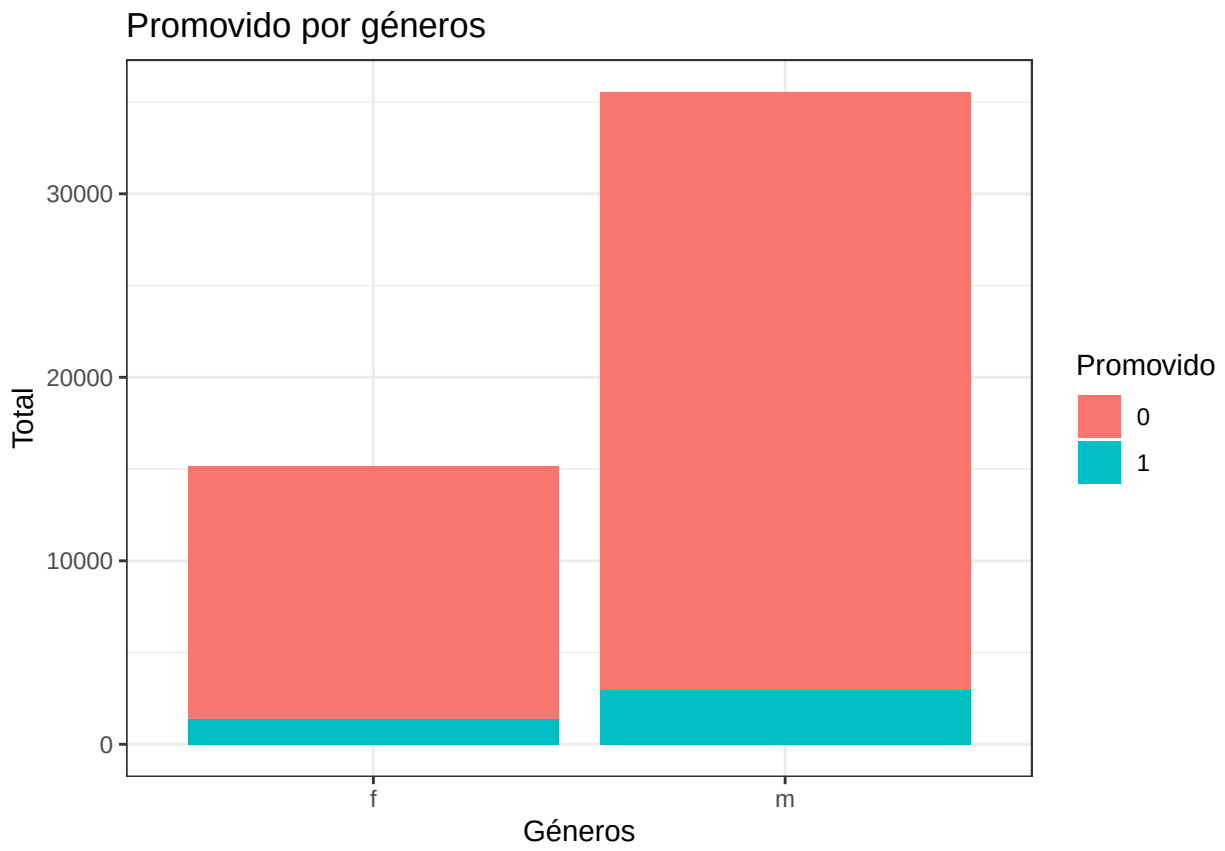
```
ggplot(data = data_p1, aes(x = department, fill = is_promoted)) +
  geom_bar(position = "stack") +
  labs(title = "Promocionado según departamento",
       x = "Departamento",
       y = "Total",
       fill = "Promovido") +
  theme_bw()
```



Con este gráfico de barras apiladas vemos que esta variable no afecta fuertemente a la promoción ya que cada departamento tiene resultados similares si no contamos en proporción.

Por último vamos a ver si el género afecta.

```
ggplot(data = data_p1, aes(x = gender, fill = is_promoted)) +
  geom_bar(position = "stack") +
  labs(title = "Promovido por géneros",
       x = "Géneros",
       y = "Total",
       fill = "Promovido") +
  theme_bw()
```

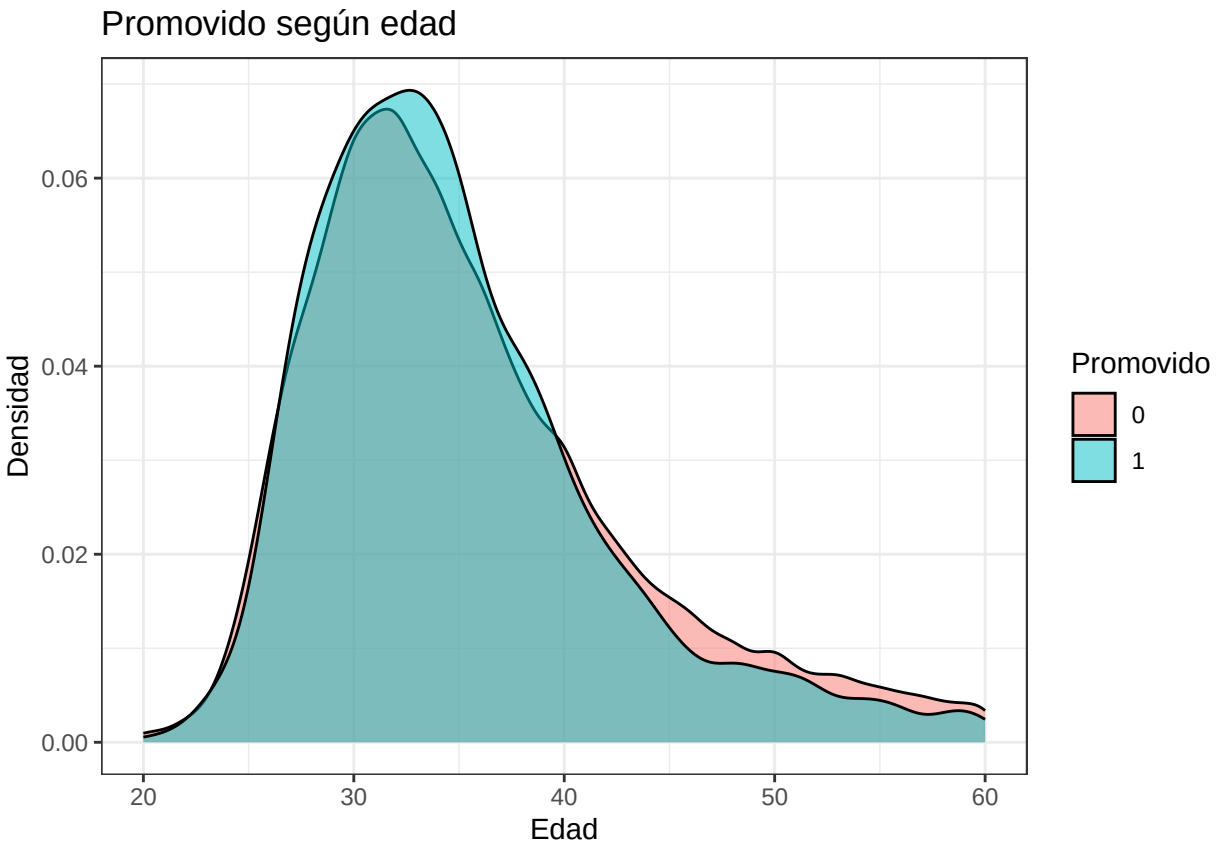


Aqui vemos que en proporcion que no afecta el género, ya que a simple vista los porcentajes de promocio segun el sexo parecen cercanos.

Ahora pasamos a las variables numéricas:

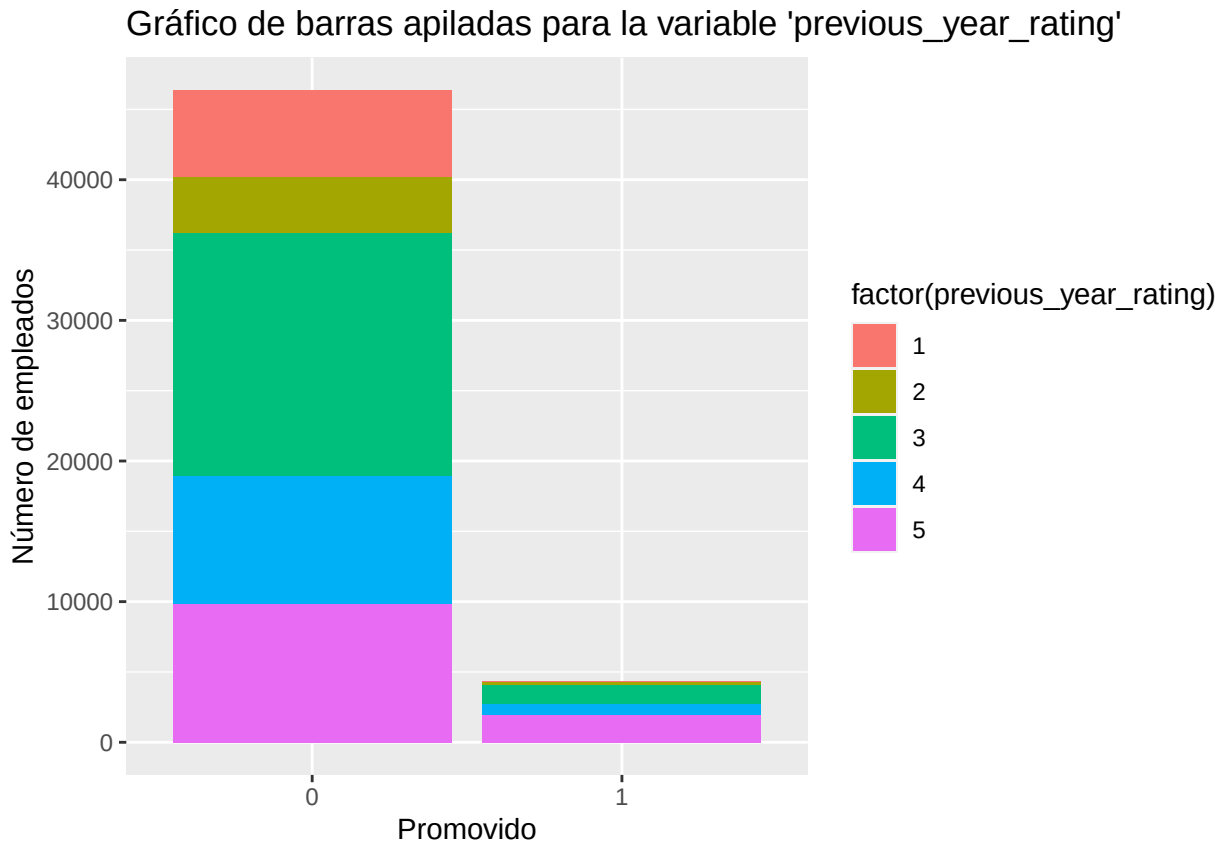
Empezamos con la edad representada en un gráfico de densidad para ver donde se concentran la mayoría de promocionados por si vemos que la edad pueda afectar ya que también está relacionada con el tiempo que lleves en la empresa

```
ggplot(data = data_p1, aes(x = age, fill = is_promoted)) +
  geom_density(alpha = 0.5) +
  labs(title = "Promovido según edad",
       x = "Edad",
       y = "Densidad",
       fill = "Promovido") +
  theme_bw()
```



El rango es bastante amplio por lo que no vemos que la edad concreta pueda afectar, pero si puede afectar dependiendo de los años que lleve en la empresa, es decir, una persona con 40 años tiene más probabilidad de llevar más tiempo en la empresa que una persona de 20 años. Por eso vamos a graficar ahora si el tiempo en la empresa afecta.

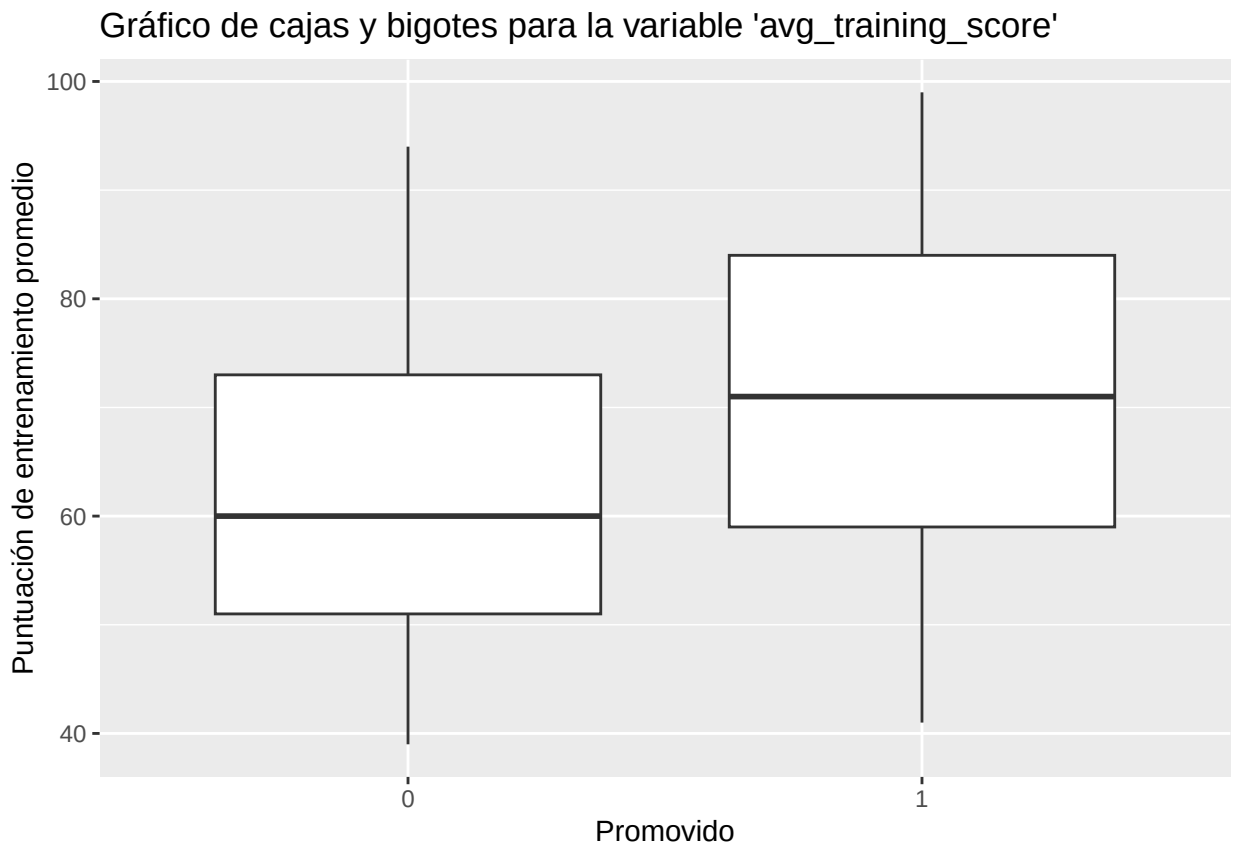
```
ggplot(data = data_p1, aes(x = is_promoted,
                           fill = factor(previous_year_rating))) +
  geom_bar(position = "stack") +
  labs(x = "Promovido", y = "Número de empleados") +
  ggtitle("Gráfico de barras apiladas para la variable 'previous_year_rating'")
```



Ahora si encontramos dos grupos que afectan a la promoción, los que llevan 5 años y 3.

Por último vamos a probar con la variable avg_training_score para ver si esta nos afecta. Para ello vamos con un gráfico de cajas y bigotes

```
ggplot(data = data_p1, aes(x = is_promoted, y = avg_training_score)) +
  geom_boxplot() +
  labs(x = "Promovido", y = "Puntuación de entrenamiento promedio") +
  ggtitle("Gráfico de cajas y bigotes para la variable 'avg_training_score'")
```



Observamos que la variable av_training_score es bastante importante ya que por encima de cierto rango se concentra todos los que han sido promovidos.

Pregunta 2:

¿Podemos averiguar qué variables afectan más a través de la predicción?

Esta pregunta se plantea responder mediante una predicción utilizando validación cruzada (cross validation), para ello podemos tirar de regresión logística o un árbol de decisión. En este caso usaremos árboles de decisión para así también ver qué variables influyen y poder comparar este resultado con otras preguntas.

Volvemos a partir de un dataset exclusivo para esta pregunta:

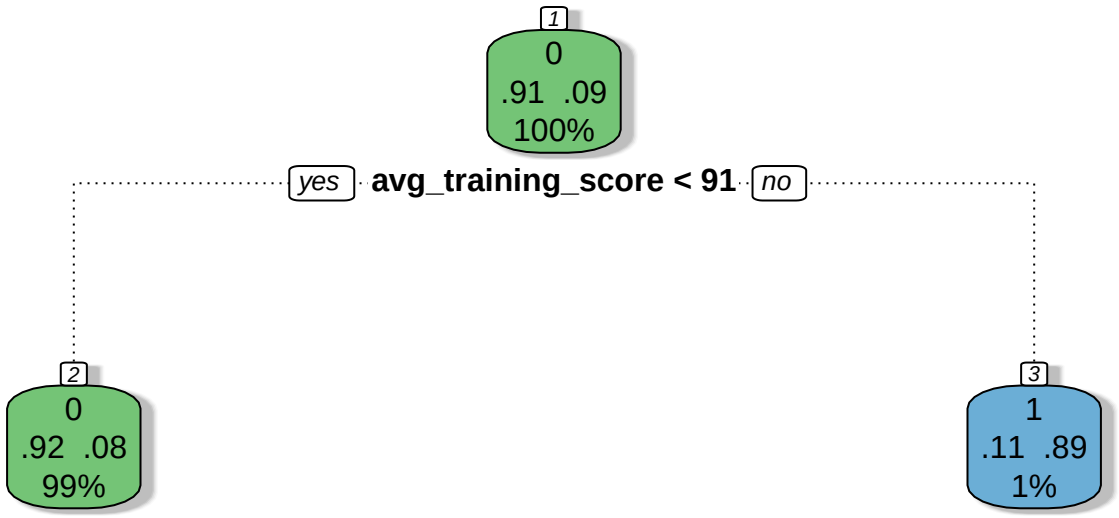
```
data_p2 <- data
```

Dividimos el dataset en conjunto de prueba(test) y entrenamiento(train) para obtener resultado

```
division <- createDataPartition(data_p2$is_promoted, p = .7, list = FALSE,
                                times = 1)
train <- data_p2[division, ]
test  <- data_p2[-division, ]
```

Entrenamos el modelo y visualizamos para tener una primera imagen de las variables decisivas

```
arbol <- rpart(formula =  is_promoted  ~ ., data = train, method = 'class')

fancyRpartPlot(arbol)
```



Podemos observar que la variable que más afecta es la puntuación en el rango superior a 91, donde con un 99% de posibilidades serás promocionado.

Pasamos a predecir según clasificación

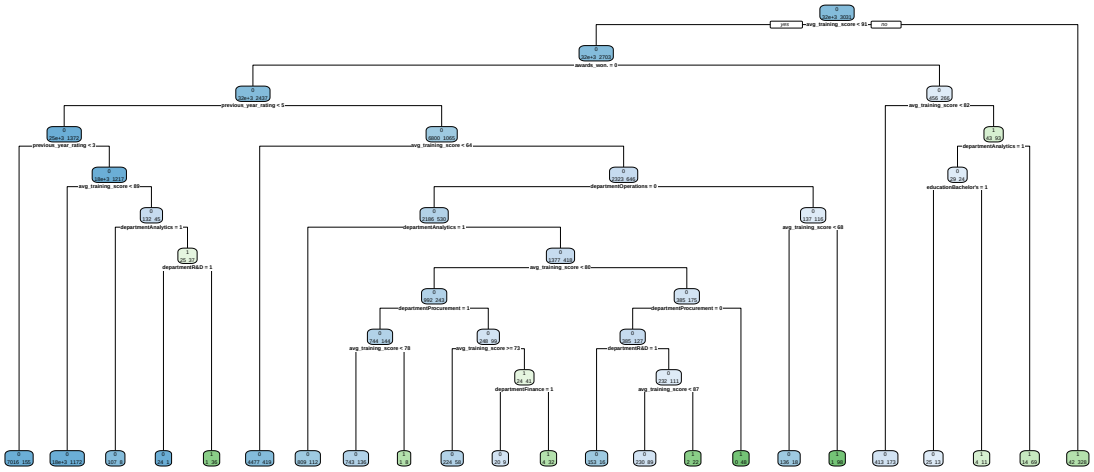
```
prediccion <-predict(arbol, test, type = "class")
```

Ahora toca evaluar el rendimiento según validación cruzada.

```
valcruz <- trainControl(method = "cv", number = 10)

arbolfit <- train(is_promoted ~ ., data = train, method = "rpart",
                  trControl = valcruz, tuneLength = 10)
```

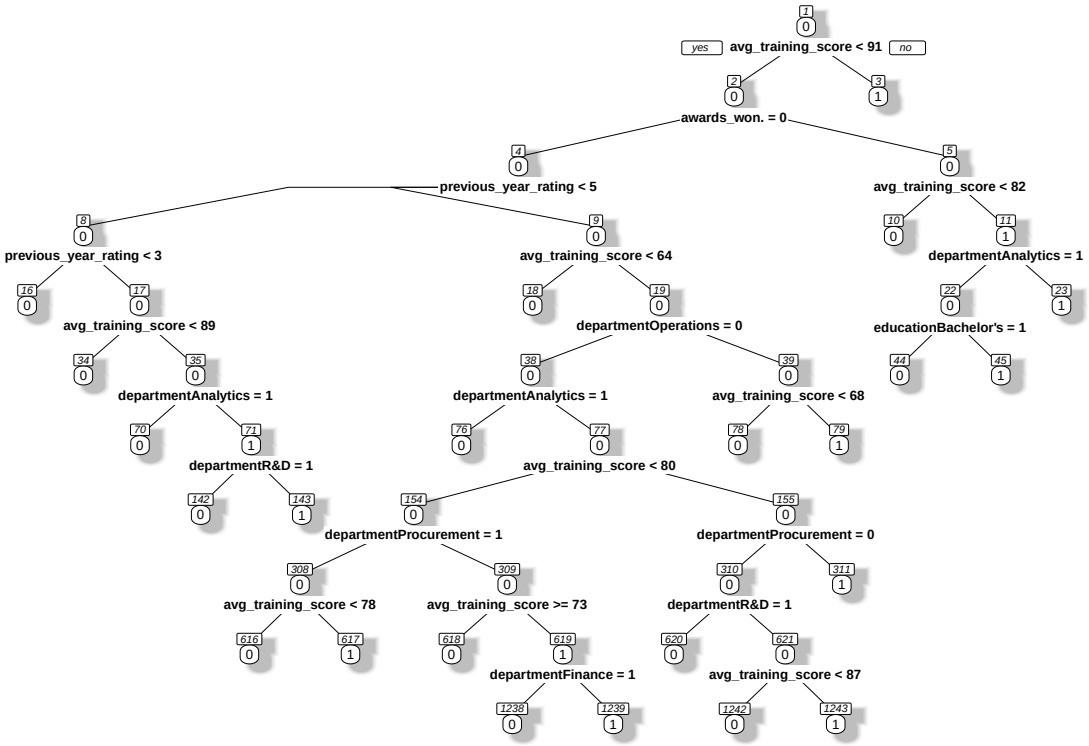
Graficamos

```
rpart.plot(arbolfit$finalModel, type = 2, extra = 1)
```



Mejoramos la visualización para ver mejor el nombre de las variable y la estructura del árbol

```
prp(arbolfit$finalModel, type = 2, nn = TRUE,
    fallen.leaves = FALSE,
    varlen = 0,  shadow.col = "gray")
```




```
prediccionTest <- predict(arbolfit, newdata = test)
confusionMatrix(prediccionTest, test$is_promoted)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction     0     1
##          0 13861  1044
##          1    45   254
##
##                Accuracy : 0.9284
##                  95% CI : (0.9242, 0.9324)
##     No Information Rate : 0.9146
##     P-Value [Acc > NIR] : 2.733e-10
##
##                   Kappa : 0.2956
##
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.9968
##             Specificity : 0.1957
##          Pos Pred Value : 0.9300
##          Neg Pred Value : 0.8495
##              Prevalence : 0.9146
##          Detection Rate : 0.9117
```

```
##    Detection Prevalence : 0.9803
##       Balanced Accuracy : 0.5962
##
##        'Positive' Class : 0
##
```

Con estos datos ya podemos sacar conclusiones sobre el modelo:

- Hay 13,861 verdaderos negativos y 254 verdaderos positivos
- Hay 1,044 falsos negativos y 45 falsos positivos

Ya con estos valores vemos que el modelo está bien ajustado. Con estos datos obtenemos una precisión del 92,8% y una tasa de error del 7,16%. También observamos que tenemos mayor capacidad para acertar, es decir predecir cuándo se va a promover que cuando no se va a promover.

Además observando el arbolfit vemos que ya aparecen más variables que son relevantes como award_won y previus_year_rating. Añadir que vemos una fuerte tendencia en el departamento de analiticas, donde superando los 82 puntos serás promocionado con alta seguridad.

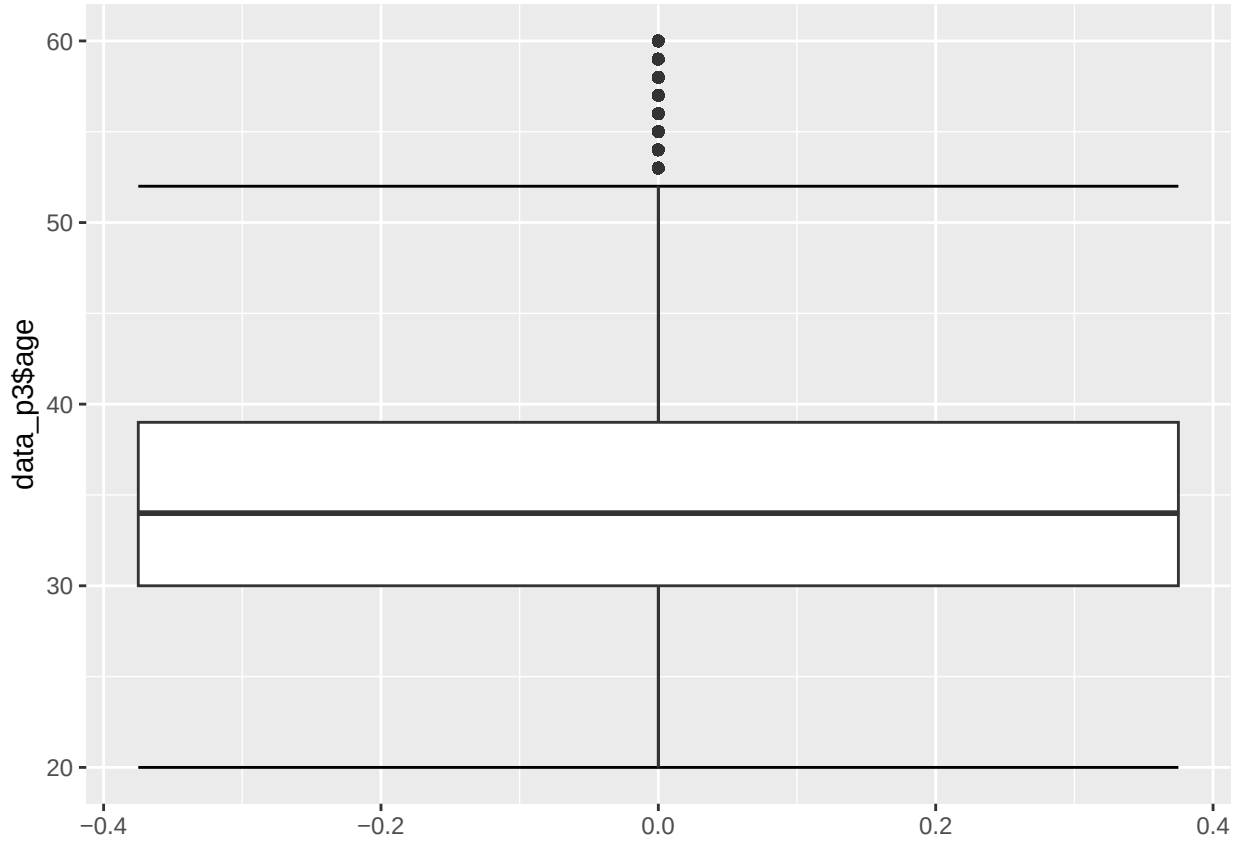
Pregunta 3:

¿Afectan los valores atípicos los resultados del modelo predictivo?

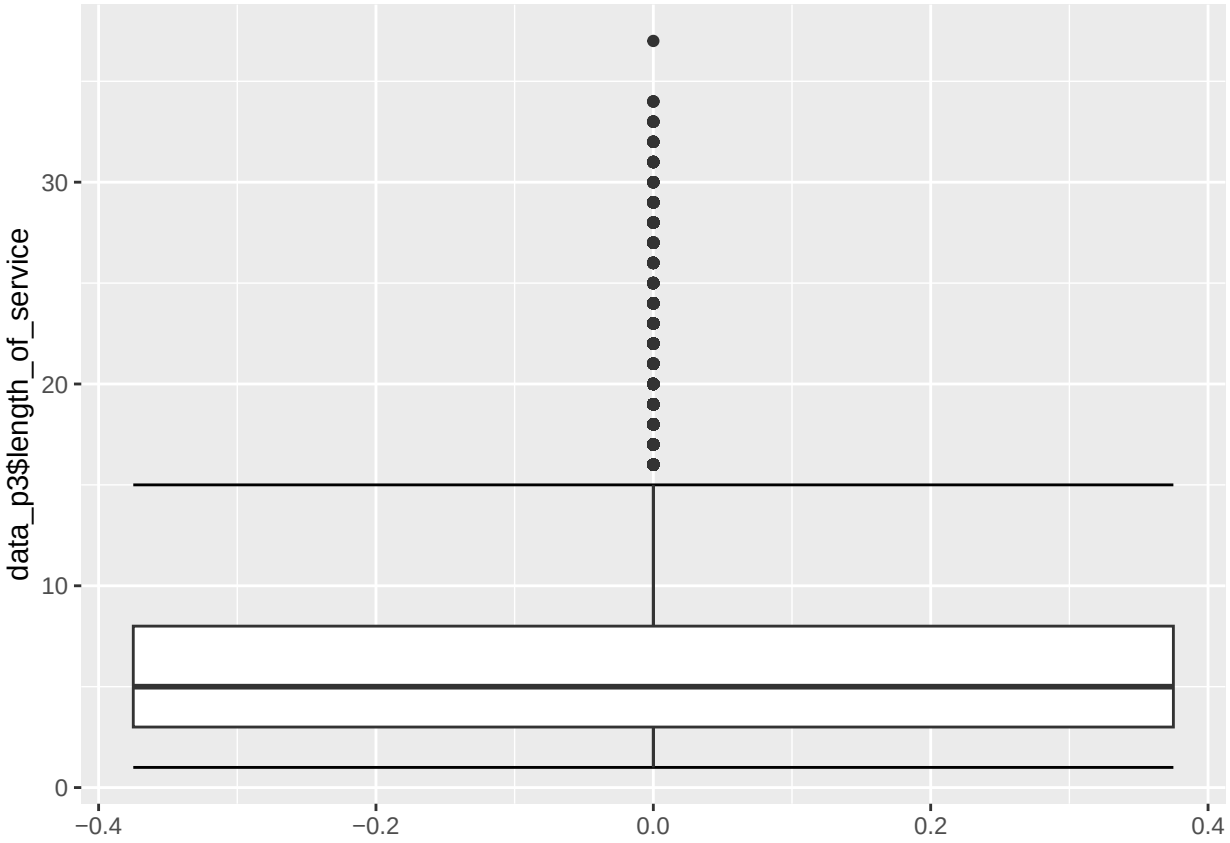
Conociendo los resultados de nuestro modelo predictivo, nos surge la duda: de que manera podríamos mejorar los resultados de nuestro modelo, nos surgió la idea de trabajar con los datos atípicos.

Procedimos a graficar mediante los gráficos de cajas los diferentes atributos del dataset para identificar cuales contenían valores atípicos.

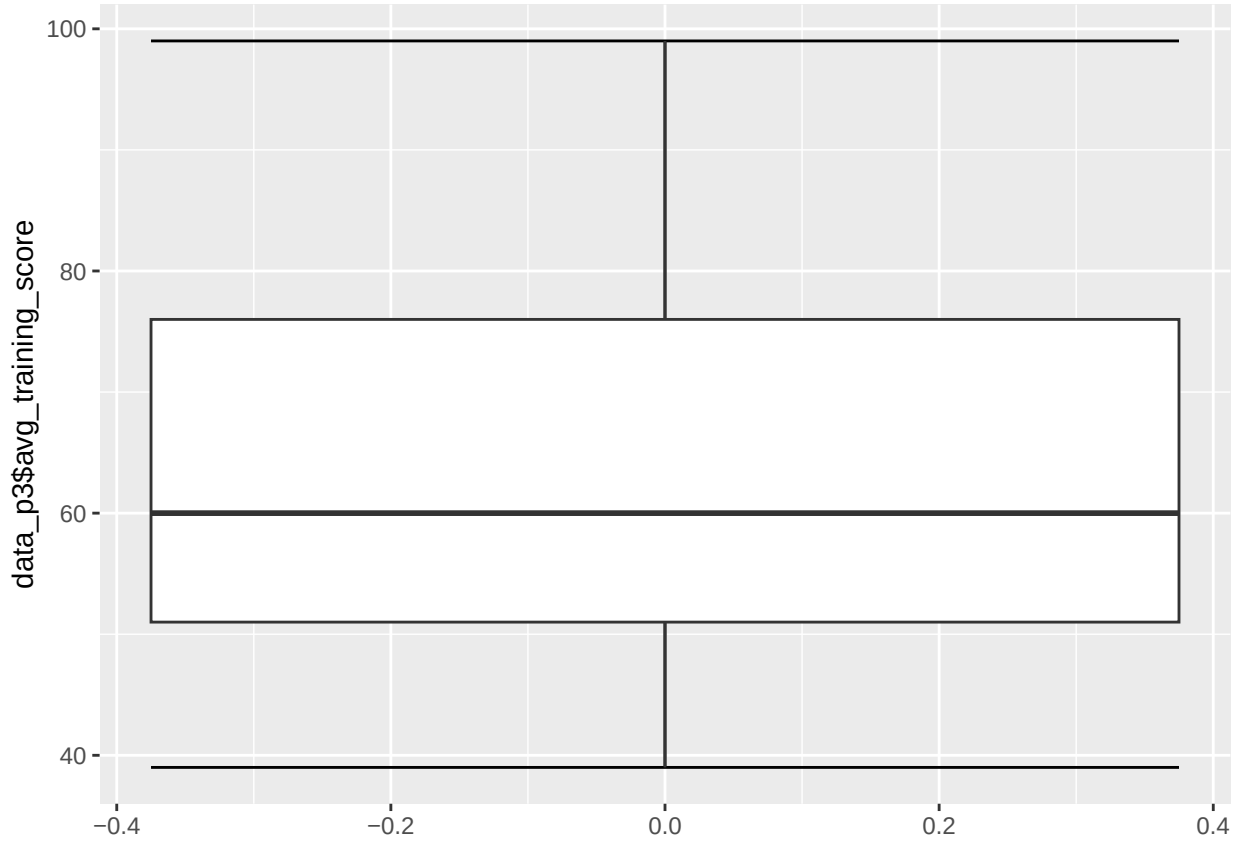
```
data_p3 <- data

ggplot(data_p3, aes( y = data_p3$age)) +
  stat_boxplot(geom ='errorbar') +
  geom_boxplot()
```



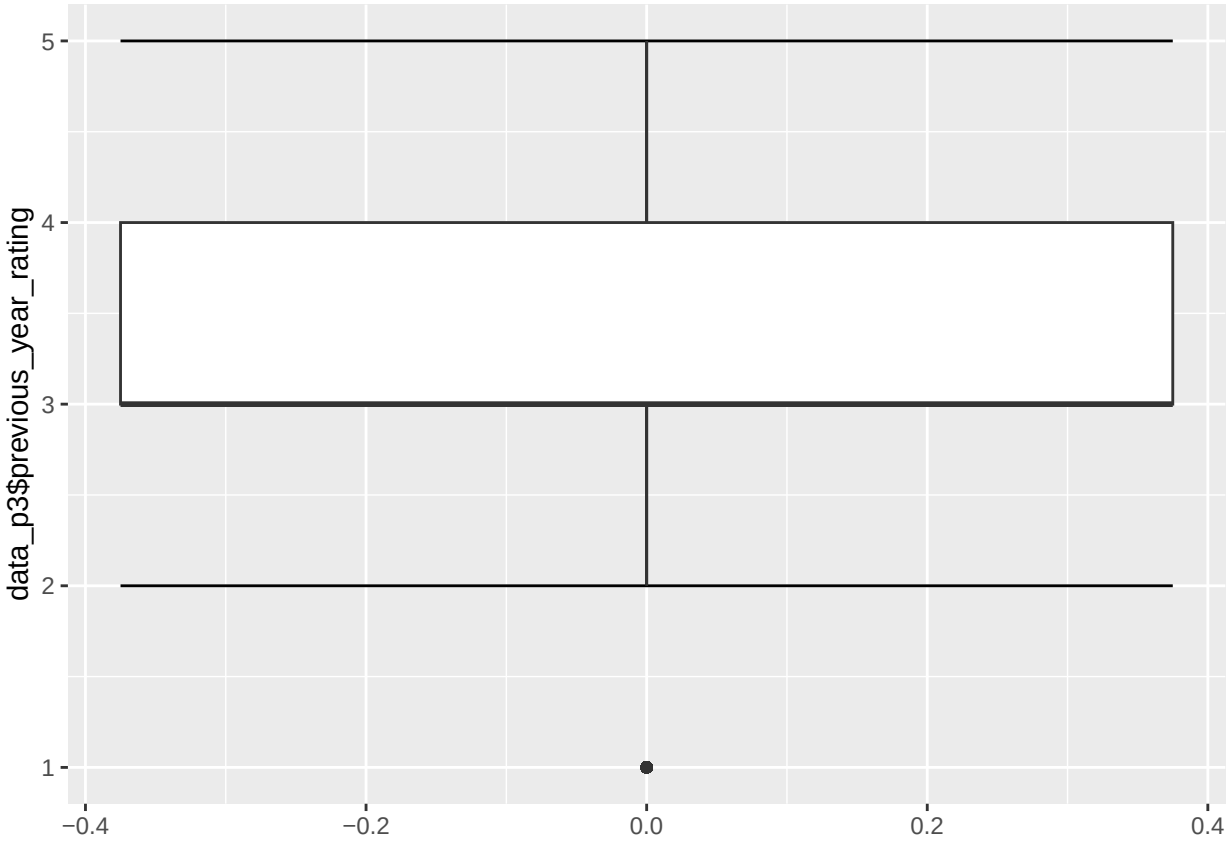

```
ggplot(data_p3, aes( y = data_p3$length_of_service)) +
  stat_boxplot(geom ='errorbar') +
  geom_boxplot()
```




```
ggplot(data_p3, aes( y = data_p3$avg_training_score)) +
  stat_boxplot(geom ='errorbar') +
  geom_boxplot()
```




```
ggplot(data_p3, aes( y = data_p3$previous_year_rating)) +
  stat_boxplot(geom ='errorbar') +
  geom_boxplot()
```



De los cuales “Edad” y “Cantidad de años de servicios” era los atributos que contenía valores atípicos o muy por encima de la media.

Realizamos el conteo de los datos que se han visualizado como atípicos

```
cantidadEdad <- sum(data_p3$age > 55)
cantidadServicio <- sum(data_p3$length_of_service > 15)

print(cantidadEdad)
```

```
## [1] 1141
```

```
print(cantidadServicio)
```

```
## [1] 2347
```

```
mensaje <- paste("Total de datos atípicos: ", cantidadEdad+ cantidadServicio)
print(mensaje)
```

```
## [1] "Total de datos atípicos:  3488"
```

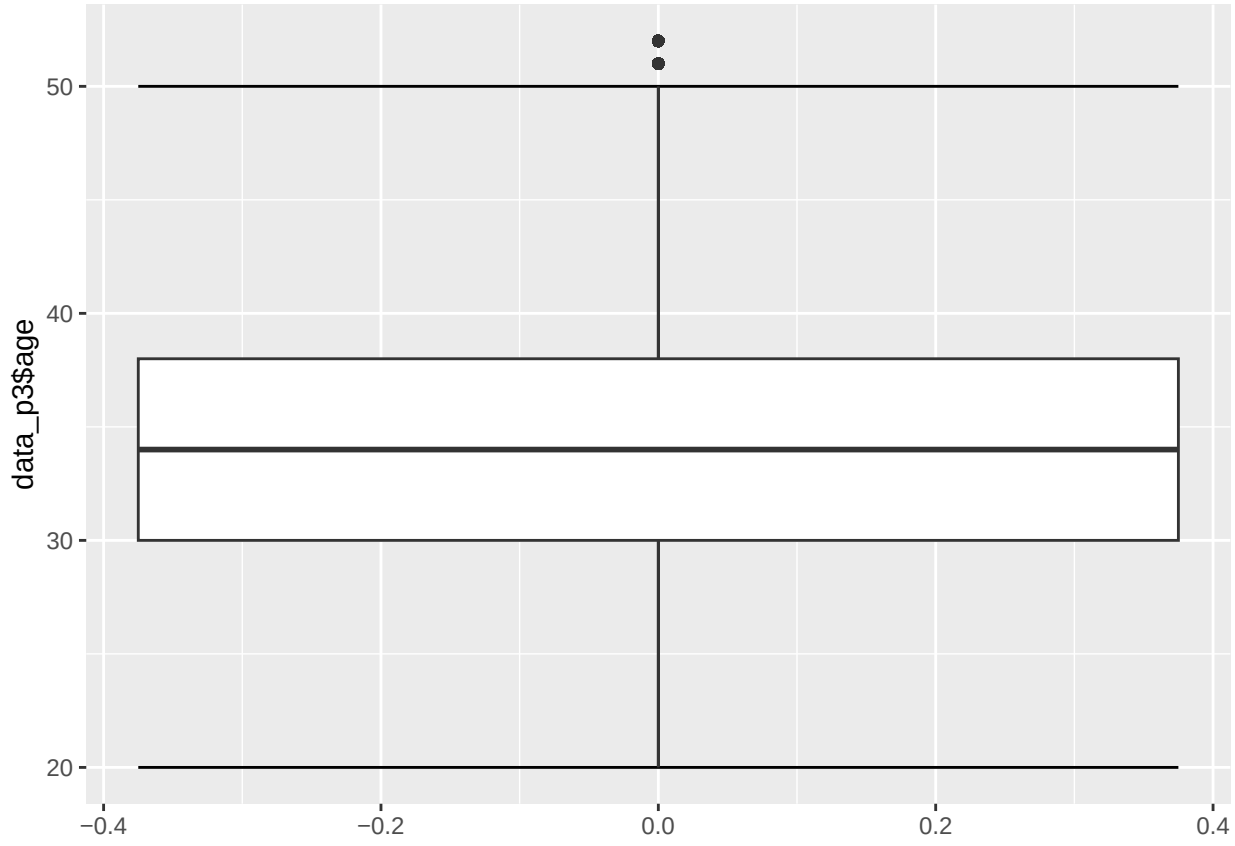
Por la cantidad de datos que se veían involucrados decidimos sustituirlo por el valor promedio de los mismos, esto para ambos atributos que hemos identificado.

```r
#Tratamiento del atributo Edad

q1 <- quantile(data_p3$age, 0.25)
q3 <- quantile(data_p3$age, 0.75)
iqr <- q3 - q1


atipicos <- data_p3$age < q1 - 1.5*iqr | data_p3$age > q3 + 1.5*iqr


media <- mean(data_p3$age[!atipicos])

data_p3$age[atipicos] <- media
```

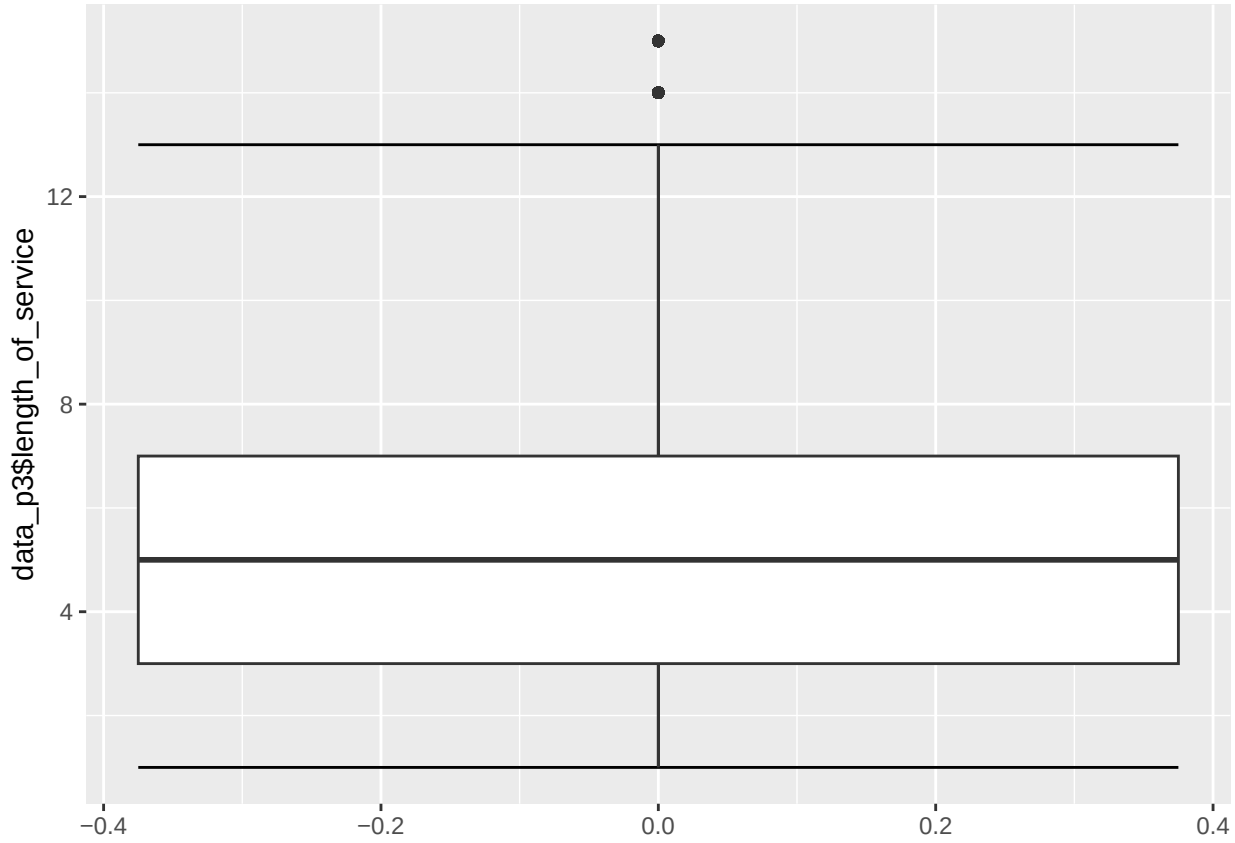
```r
#Tratamiento del atributo Cantidad de Servicio

q1 <- quantile(data_p3$length_of_service, 0.25)
q3 <- quantile(data_p3$length_of_service, 0.75)
iqr <- q3 - q1


atipicos <- data_p3$length_of_service < q1 - 1.5*iqr|
  data_p3$length_of_service > q3 + 1.5*iqr


media <- mean(data_p3$length_of_service[!atipicos])

data_p3$length_of_service[atipicos] <- media
```

Luego de haber sustituido los valores atípicos, realizamos los diagramas de cajas para ver la media.

```r
ggplot(data_p3, aes( y = data_p3$age)) +
  stat_boxplot(geom ='errorbar') +
  geom_boxplot()
```




```
ggplot(data_p3, aes( y = data_p3$length_of_service)) +
  stat_boxplot(geom ='errorbar') +
  geom_boxplot()
```



Procedemos a implementar nuevamente el modelo predictivo que utilizamos al inicio, pero utilizado el nuevo dataset.

```
#División del nuevo dataset en train y test

division2 <- createDataPartition(data_p3$is_promoted, p = .7, list = FALSE,
                                 times = 1)
train2 <- data_p3[division, ]
test2  <- data_p3[-division, ]
```

```
#Entremiento del modelo

valcruz2 <- trainControl(method = "cv", number = 10)

arbolfit2 <- train(is_promoted ~ ., data = train2, method = "rpart",
                   trControl = valcruz2, tuneLength = 10)
```

```
#Creación de matriz de confusión

prediccionTest2 <- predict(arbolfit2, newdata = test2)
confusionMatrix(table(prediccionTest2, test2$is_promoted))
```

```
## Confusion Matrix and Statistics
##
##
```

```
## prediccionTest2     0     1
##               0 13843   965
##               1    63   333
##
##                Accuracy : 0.9324
##                  95% CI : (0.9283, 0.9363)
##     No Information Rate : 0.9146
##     P-Value [Acc > NIR] : 3.041e-16
##
##                   Kappa : 0.3679
##
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.9955
##             Specificity : 0.2565
##          Pos Pred Value : 0.9348
##          Neg Pred Value : 0.8409
##              Prevalence : 0.9146
##          Detection Rate : 0.9105
##    Detection Prevalence : 0.9740
##       Balanced Accuracy : 0.6260
##
##        'Positive' Class : 0
##
```

En vista de los resultados obtenidos, podemos notar que tenemos un accuracy el 93.24 en esta nueva predicción, frente a ala anterior que fue del 92.84, por lo que hemos tenido una mejora de 0.40.

En conclusión, Hemos obtenido una muy peque mejora en el entrenamiento de este nuevo modelo, lo que nos da como resultado que los valores atípicos, o valores por fuera de la media, no afectan potencialmente la predicción.