Becas Crema

Eric Bellet

11 de marzo de 2016

Introducción

El objetivo del script **usage.R** es utilizar el API de **Google Maps Distance Matrix API** con la finalidad de obtener la distancia y el tiempo entre una dirección de un apartamento hacia la **Universidad Sapienza** de Roma para utilizarlas como criterio de selección de un hogar adecuado para un estudiante. Posterior al uso del API, se realizará un **preprocesamiento** general del dataset para poder aplicar el algoritmo de aprendizaje supervisado **regresión lineal**.

Carga del set de datos y utilización del API

Utilizo mi propia key para realizar peticiones al API donde la función **parse_data** me devuelve la distancia y el tiempo entre la dirección de un apartamento y la **Universidad Sapienza**.

```
source("C:/Users/EricBellet/Desktop/AprendizajeSupervisado/src/Escogiendo un Hogar/google_api.R")
df <- read_excel("C:/Users/EricBellet/Desktop/AprendizajeSupervisado/data/hogares.xlsx")
df <- na.omit(df)
df$Foto <- NULL

#Inicializamos dataframe y vectores.
dataframe <- data.frame()
Distancia <- vector()
Minutos <- vector()
ori <- vector()

destino =c("Piazzale Aldo Moro")
#Realizo un ciclo para hacer peticiones al API dirección por dirección.
for (origen in df$Dirección){
#Elimino los \n
origen <- strsplit(as.character(origen), "\n")

#Utilizo mi API key
api_key = "AIzaSyDO4qKCMM18-iQzY6QK1MSmmii_aVhqUPE"
api_url = get_url(origen, destino, api_key)
datos = get_data(api_url)
#Parseo los datos obtenidos.
timedistancia = parse_data(datos)

#Concateno las distancia y el tiempo que arroja el API.
Distancia <- c(Distancia,timedistancia[1])
Minutos <- c(Minutos,timedistancia[2])
ori <- c(ori, origen)
Distancia <- cbind(Distancia)
Minutos <- cbind(Minutos)
#Guardo todos los datos parseados en un dataframe.
dataframe <- cbind(ori,Distancia,Minutos)
}#endfor
```

Transformo el tiempo en un formato único, minutos, y guardo toda la información que me dio el API en un dataframe.

```
dataframe <- as.data.frame(dataframe)
#Tranformamos todos los tiempos a minutos
enHoras <- grepl("h",dataframe$Minutos)
for (i in 1:length(enHoras)){
  if (enHoras[i] == TRUE){
  num <-  as.numeric(unlist(strsplit(unlist(dataframe$Minutos[i]), "[^0-9]+")))
  dataframe$Minutos[i] <- (num[1]*60) + num[2]
  }else{
    num <-  as.numeric(unlist(strsplit(unlist(dataframe$Minutos[i]), "[^0-9]+")))
    dataframe$Minutos[i] <- num[1]
  }#endif
}#endfor
#Agrego las columnas distancia y tiempo al dataframe que se utilizara para la regresion lineal.
df$Distancia <- dataframe$Distancia
df$Minutos <- dataframe$Minutos
#Elimino las filas cuya direccion el API no encontro.
df <- df[!df$Minutos == "NA", ]
#Asigno un valor de importancia a los tiempos
df$Minutos <- as.numeric(df$Minutos)
df["ValorMinutos"] <- as.factor(ordered(cut(df$Minutos, c(-Inf,15,60,120,180,240,300,360,420,600,Inf)),l

#Realizo una transformacion a la columna valor minutos para poder utilizarla.

for (i in 1:nrow(df)){
  df$ValorMinutos2[i] <- as.numeric(as.character(df["ValorMinutos"][[1]][i]))

}
```

Aquellos apartamentos que poseen diversas habitaciones en oferta, coloco cada una en filas distintas, para evaluarlas independientementes.

```
#AGREGO FILAS POR CADA HABITACION DISPONIBLE CON SU CORRESPONDIENTE PRECIO.

df$Disponibles[grepl("1 singola", df$`Habitaciones Disponibles`)] <- "1"
df$Disponibles[grepl("1 singole", df$`Habitaciones Disponibles`)] <- "1"
df$Disponibles[grepl("1 Singola", df$`Habitaciones Disponibles`)] <- "1"
df$Disponibles[grepl("1 Singola", df$`Habitaciones Disponibles`)] <- "1"
df$Disponibles[grepl("Intero Appartamento", df$`Habitaciones Disponibles`)] <- "1"
df$Disponibles[grepl("Intero appartamento", df$`Habitaciones Disponibles`)] <- "1"
df$Disponibles[grepl("Mini Appartamento", df$`Habitaciones Disponibles`)] <- "1"
df$Disponibles[grepl("intero appartamento", df$`Habitaciones Disponibles`)] <- "1"
df$Disponibles[grepl("Mini Appartamento", df$`Habitaciones Disponibles`)] <- "1"

df$Disponibles[grepl("2 singola", df$`Habitaciones Disponibles`)] <- "2"
df$Disponibles[grepl("2 singole", df$`Habitaciones Disponibles`)] <- "2"
df$Disponibles[grepl("2 Singola", df$`Habitaciones Disponibles`)] <- "2"
df$Disponibles[grepl("2 Singola", df$`Habitaciones Disponibles`)] <- "2"

df$Disponibles[grepl("3 singola", df$`Habitaciones Disponibles`)] <- "3"
df$Disponibles[grepl("3 singole", df$`Habitaciones Disponibles`)] <- "3"
df$Disponibles[grepl("3 Singola", df$`Habitaciones Disponibles`)] <- "3"
```

```
df$Disponibles[grepl("3 Singola", df$`Habitaciones Disponibles`)] <- "3"

df$Disponibles[grepl("4 singola", df$`Habitaciones Disponibles`)] <- "4"
df$Disponibles[grepl("4 singole", df$`Habitaciones Disponibles`)] <- "4"
df$Disponibles[grepl("4 Singola", df$`Habitaciones Disponibles`)] <- "4"
df$Disponibles[grepl("4 Singola", df$`Habitaciones Disponibles`)] <- "4"

df$Disponibles[is.na(df$Disponibles)] <- 1

#Replico las filas que posee mas de una habitacion disponible.
df <- df[rep(seq_len(nrow(df)), df$Disponibles),]
```

Posteriormente asigno el precio correspondiente a cada habitación.

```
#Asigno precio a cada habitacion.
i <- 1
while (i != (nrow(df)+1)) {
  array <- na.omit(as.numeric(unlist(strsplit(unlist(df$`Precio Mensual`[i]),
                                              "[^0-9]+"))))
 if (df$Disponibles[i] == 1){
   df$PrecioTotal[i] <- array[1]
   i <- i + 1
 }else{
   for (j in 1:length(array)) {
    df$PrecioTotal[i + (j-1)] <- array[j]
   }
   i <- i + as.numeric(df$Disponibles[i])
 }

}#endwhile
```

Genero 2 nuevas columnas, una donde categorizo si el precio de la habitación tiene todo incluido o no, **TodoIncluido** y otra columna que representa un valor o puntación que le asigne desde mi punto personal, **100** en caso de que este todo incluido y **0** en caso contrario. Estos valores representan el valor personal de la habitación donde mas alto sea mayor importancia tiene.

```
#Creo una columna donde coloco el valor si esta todo incluido o no.
df$TuttoIncluido[grepl("TUTTO INCLUSO", df$`Precio Mensual`)] <- 100
df$TuttoIncluido[grepl("Tutto incluso", df$`Precio Mensual`)] <- 100
df$TuttoIncluido[grepl("tutto incluso", df$`Precio Mensual`)] <- 100
df$TuttoIncluido[is.na(df$TuttoIncluido)] <- 0

#Columna a utilizar para la regresion lineal donde 1 es si tiene todo incluido y 0 es no
df$TodoIncluido[grepl("TUTTO INCLUSO", df$`Precio Mensual`)] <- 1
df$TodoIncluido[grepl("Tutto incluso", df$`Precio Mensual`)] <- 1
df$TodoIncluido[grepl("tutto incluso", df$`Precio Mensual`)] <- 1
df$TodoIncluido[is.na(df$TodoIncluido)] <- 0
```

Posteriormente creo diferentes columnas a partir de la columna Descripción.

Pasillo: Una columna que indica cuantos hay en el apartamento. Cocina: Una columna que indica cuantos hay en el apartamento. *Cuarto: Una columna que indica cuantos hay en el apartamento.* Bagno: Una columna que indica cuantos hay en el apartamento. *Balcon: Una columna que indica cuantos hay en el*

apartamento. Comedor: Una columna que indica cuantos hay en el apartamento. *Armario: Una columna que indica cuantos hay en el apartamento.* Salon: Una columna que indica cuantos hay en el apartamento. *Descripción3: Una columna que indica un valor personal a cada componente del apartamento.

```
#Etiqueto la columna descripcion donde separo por coma y e (solo el conector).
separador <- function(x)
  splat <- unlist(strsplit(x, ", | e "))

df$Descripción2 <- lapply(df$Descripción, separador)
x <- vector()
#Genero columnas que utilizare en la regresion lineal
#Descripcion3 es una columna que genera valor segun los componente de la habitacion.
df$Descripción3 <- 0
df$Pasillo <- 0
df$Cocina <- 0
df$Cuarto <- 0
df$Bagno <- 0
df$Balcon <- 0
df$Comedor <- 0
df$Armario <- 0
df$Salon <- 0
#Obtengo cuantas y cuales habitaciones tiene el apartamento y genero valor.
for (i in 1:nrow(df)) {
   for (j in 1:length(unlist(df$Descripción2[i]))) {

    x[1] <- as.numeric(unlist(strsplit(unlist(df$Descripción2[i])[j],
                              "[^0-9]+")))

    if (is.na(x) == TRUE){
      x[1] <- 1
    }

    if (grepl("Ingresso", unlist(df$Descripción2[i])[j]) == TRUE){
      df$Descripción3[i] <- df$Descripción3[i] + (5 * x[1])
      df$Pasillo[i] <- df$Pasillo[i] +x[1]
    }
    if (grepl("ingresso", unlist(df$Descripción2[i])[j]) == TRUE){
      df$Descripción3[i] <- df$Descripción3[i] + (5 * x[1])
      df$Pasillo[i] <- df$Pasillo[i] +x[1]
    }

    if (grepl("cucina", unlist(df$Descripción2[i])[j]) == TRUE){
      df$Descripción3[i] <- df$Descripción3[i] + (20 * x[1])
      df$Cocina[i] <- df$Cocina[i] +x[1]
    }

    if (grepl("angolo cottura", unlist(df$Descripción2[i])[j]) == TRUE){
      df$Descripción3[i] <- df$Descripción3[i] + (20 * x[1])
      df$Cocina[i] <- df$Cocina[i] +x[1]
    }

    if (grepl("stanze", unlist(df$Descripción2[i])[j]) == TRUE){
      df$Descripción3[i] <- df$Descripción3[i] + (10 * x[1])
      df$Cuarto[i] <- df$Cuarto[i] +x[1]
```

```
  }

  if (grepl("camere", unlist(df$Descripción2[i])[j]) == TRUE){
    df$Descripción3[i] <- df$Descripción3[i] + (10 * x[1])
    df$Cuarto[i] <- df$Cuarto[i] +x[1]
  }

  if (grepl("camera", unlist(df$Descripción2[i])[j]) == TRUE){
    df$Descripción3[i] <- df$Descripción3[i] + (10 * x[1])
    df$Cuarto[i] <- df$Cuarto[i] +x[1]
  }

  if (grepl("bagni", unlist(df$Descripción2[i])[j]) == TRUE){
    df$Descripción3[i] <- df$Descripción3[i] + (15 * x[1])
    df$Bagno[i] <- df$Bagno[i] +x[1]
  }

  if (grepl("bagno", unlist(df$Descripción2[i])[j]) == TRUE){
    df$Descripción3[i] <- df$Descripción3[i] + (15 * x[1])
    df$Bagno[i] <- df$Bagno[i] +x[1]
  }

  if (grepl("disimpegno", unlist(df$Descripción2[i])[j]) == TRUE){
    df$Descripción3[i] <- df$Descripción3[i] + (5 * x[1])
  }

  if (grepl("balcone", unlist(df$Descripción2[i])[j]) == TRUE){
    df$Descripción3[i] <- df$Descripción3[i] + (5 * x[1])
    df$Balcon[i] <- df$Balcon[i] +x[1]
  }

  if (grepl("ampiio terrazzo", unlist(df$Descripción2[i])[j]) == TRUE){
    df$Descripción3[i] <- df$Descripción3[i] + (5 * x[1])
    df$Balcon[i] <- df$Balcon[i] +x[1]
  }

  if (grepl("sala da pranzo", unlist(df$Descripción2[i])[j]) == TRUE){
    df$Descripción3[i] <- df$Descripción3[i] + (30 * x[1])
    df$Comedor[i] <- df$Comedor[i] +x[1]
  }

  if (grepl("doppio soggiorno", unlist(df$Descripción2[i])[j]) == TRUE){
    df$Descripción3[i] <- df$Descripción3[i] + (20 * x[1])
    df$Salon[i] <- df$Salon[i] +x[1]
  }

  if (grepl("salotto", unlist(df$Descripción2[i])[j]) == TRUE){
    df$Descripción3[i] <- df$Descripción3[i] + (20 * x[1])
    df$Salon[i] <- df$Salon[i] +x[1]
  }

  if (grepl("armario", unlist(df$Descripción2[i])[j]) == TRUE){
    df$Descripción3[i] <- df$Descripción3[i] + (5 * x[1])
```

```
      df$Armario[i] <- df$Armario[i] +x[1]
    }

    if (grepl("ripostiglio", unlist(df$Descripción2[i])[j]) == TRUE){
      df$Descripción3[i] <- df$Descripción3[i] + (5 * x[1])
      df$Armario[i] <- df$Armario[i] +x[1]
    }

    if (grepl("corridoio", unlist(df$Descripción2[i])[j]) == TRUE){
      df$Descripción3[i] <- df$Descripción3[i] + (5 * x[1])
      df$Pasillo[i] <- df$Pasillo[i] +x[1]
    }

    if (grepl("Appartamento su due livelli", unlist(df$Descripción2[i])[j]) == TRUE){
      df$Descripción3[i] <- df$Descripción3[i] + (50 * x[1])
    }
  }#endfor
}#endfor
```

```
## Warning in x[1] <- as.numeric(unlist(strsplit(unlist(df$Descripción2[i])
## [j], : número de items para para sustituir no es un múltiplo de la longitud
## del reemplazo

## Warning in x[1] <- as.numeric(unlist(strsplit(unlist(df$Descripción2[i])
## [j], : número de items para para sustituir no es un múltiplo de la longitud
## del reemplazo

## Warning in x[1] <- as.numeric(unlist(strsplit(unlist(df$Descripción2[i])
## [j], : número de items para para sustituir no es un múltiplo de la longitud
## del reemplazo

## Warning in x[1] <- as.numeric(unlist(strsplit(unlist(df$Descripción2[i])
## [j], : número de items para para sustituir no es un múltiplo de la longitud
## del reemplazo

## Warning in x[1] <- as.numeric(unlist(strsplit(unlist(df$Descripción2[i])
## [j], : número de items para para sustituir no es un múltiplo de la longitud
## del reemplazo

## Warning in x[1] <- as.numeric(unlist(strsplit(unlist(df$Descripción2[i])
## [j], : número de items para para sustituir no es un múltiplo de la longitud
## del reemplazo

## Warning in x[1] <- as.numeric(unlist(strsplit(unlist(df$Descripción2[i])
## [j], : número de items para para sustituir no es un múltiplo de la longitud
## del reemplazo
```

```
#MUESTRA UN WARNING QUE EN REALIDAD NO CAUSA NINGUN PROBLEMA
```

Creo una columna llamada TipoHabitacion donde se encuentran las categorias de las habitaciones disponibles, **Intero Appartamento**, **monolocale**, **singola**, entre otros.

```
#Categorizo los tipos de habitacion.
df$TipoHabitacion[grepl("Intero Appartamento", df$`Habitaciones Disponibles`)] <- "1"
df$TipoHabitacion[grepl("Intero appartamento", df$`Habitaciones Disponibles`)] <- "1"
df$TipoHabitacion[grepl("intero appartamento", df$`Habitaciones Disponibles`)] <- "1"

df$TipoHabitacion[grepl("monolocale", df$`Habitaciones Disponibles`)] <- "2"
df$TipoHabitacion[grepl("Mini Appartamento", df$`Habitaciones Disponibles`)] <- "2"

df$TipoHabitacion[grepl("posto letto", df$`Habitaciones Disponibles`)] <- "4"
df$TipoHabitacion[grepl("doppia", df$`Habitaciones Disponibles`)] <- "5"
df$TipoHabitacion[grepl("doppie", df$`Habitaciones Disponibles`)] <- "5"

df$TipoHabitacion[grepl("singola", df$`Habitaciones Disponibles`)] <- "3"
df$TipoHabitacion[grepl("singole", df$`Habitaciones Disponibles`)] <- "3"
df$TipoHabitacion[grepl("Singola", df$`Habitaciones Disponibles`)] <- "3"
df$TipoHabitacion[grepl("Singole", df$`Habitaciones Disponibles`)] <- "3"
```

Finalmente utilizo la columna valorTotal, que representa valores del 1 al 10 donde 10 es la ponderación más alta del inmueble. El valortotal es igual al valor en tiempo (donde menor sea el tiempo de la residencia a la Universidad el valor es más alto), mas el valor del inmueble (cuantos baños tiene, si tiene cocina, balcon, etc), mas el valor de si el precio tiene todo incluido.

```
#-------------------Cual es la mejor habitacion-------------------------
#df$Disponibles <- NULL
#df$Descripción2 <- NULL
df$valorInmobiliario <- df$Descripción3
#df$Descripción3 <- NULL

#El valor de un apto esta relacionado con el tiempo, cuantas habitaciones posee
# y si el precio tiene todo incluido

df$ValorTotal <- df$ValorMinutos2 + df$valorInmobiliario + df$TuttoIncluido

df$ValorMinutos <- NULL
#df$ValorMinutos2 <- NULL
#df$TuttoIncluido <- NULL
#df$valorInmobiliario <- NULL

#Escalo o estandarizo el valor entre 1 y 10.
range01 <- function(x){((x-min(x))/(max(x)-min(x)))*10}
df$ValorTotal <- range01(df$ValorTotal)
df$TipoHabitacion <- as.numeric(df$TipoHabitacion)
```

Separación por sexo

Creo 2 nuevos dataframes uno para mujeres y otro para hombres para aplicar regresión lineal a cada uno, ya que es posible que el precio sea afectado por el sexo por lo tanto es mejor evaluarlos por separado.

```
df$Disponibles <- as.numeric(df$Disponibles)
df$TipoHabitacion <- as.numeric(df$TipoHabitacion)
#DIVIDO EL DATAFRAME EN DOS, MUJERES Y HOMBRES.
df$Sexo <- df$Notas
```

```
df$Sexo[grepl("ragazzi/e", df$Sexo)] <- "Ambos"
df$Sexo[grepl("ragazze/i", df$Sexo)] <- "Ambos"
df$Sexo[grepl("ragazzi/ragazze", df$Sexo)] <- "Ambos"
df$Sexo[grepl("ragazze", df$Sexo)] <- "Femenino"
df$Sexo[grepl("ragazzi", df$Sexo)] <- "Masculino"
df$Sexo[!grepl("Ambos", df$Sexo) & !grepl("Masculino",
                                          df$Sexo) & !grepl("Femenino", df$Sexo)] <- "Ambos"
dfM <- df[df$Sexo == 'Masculino' | df$Sexo == 'Ambos',]
dfF <- df[df$Sexo == 'Femenino' | df$Sexo == 'Ambos',]
#-------------------------------------------------------------
dfM$Sexo <- NULL
dfF$Sexo <- NULL
dfM$Notas <- NULL
dfF$Notas <- NULL
```

Training y testing data

Genero un training y testing estratificado para ambos dataframes.

```
#*****************************************************************************************
#-------------------------SAMPLING PARA MASCULINO---------------------------------
#*****************************************************************************************
#Obtengo los valore unicos de mIngreso para los hombres.
valores <- unique(dfM$PrecioTotal)
totalvalores <- nrow(dfM)
probabilidad <- vector()
#Calculo la probabilidad de cada valor de mIngreso.
for (i in 1:length(valores)){
  probabilidad <- c(probabilidad, sum(dfM$PrecioTotal == valores[i]) / totalvalores)
}
asignarProb <- function(x){
  for (i in 1:length(valores)) {
    if (valores[i] == x){
      return(probabilidad[i])
    }

  }
}
#Obtengo un vector de probabilidades para cada valor de PrecioTotal
probabilidades <- lapply(dfM$PrecioTotal, asignarProb)
probabilidades<-unlist(probabilidades)
#*****************************************************************************************
#----------Genero un train y test data estratificado para hombres-------------------
#*****************************************************************************************
set.seed(1)
sets <- sample(nrow(dfM), nrow(dfM)*0.8, prob=probabilidades, replace=F)
trainingM <- dfM[sets,]
testingM <- dfM[-sets,]
#*****************************************************************************************
#-------------------------SAMPLING PARA FEMENINO---------------------------------
#*****************************************************************************************
#Obtengo los valore unicos de mIngreso para mujeres
valores <- unique(dfF$PrecioTotal)
```

```
totalvalores <- nrow(dfF)
probabilidad <- vector()
#Calculo la probabilidad de cada valor de mIngreso.
for (i in 1:length(valores)){
  probabilidad <- c(probabilidad, sum(dfF$PrecioTotal == valores[i]) / totalvalores)
}
asignarProb <- function(x){
  for (i in 1:length(valores)) {
    if (valores[i] == x){
      return(probabilidad[i])
    }

  }
}
#Obtengo un vector de probabilidades para cada valor de PrecioTotal
probabilidades <- lapply(dfF$PrecioTotal, asignarProb)
probabilidades<-unlist(probabilidades)
#*********************************************************************************
#----------Genero un train y test data estratificado para mujeres-----------------
#*********************************************************************************
set.seed(1)
sets <- sample(nrow(dfF), nrow(dfF)*0.8, prob=probabilidades, replace=F)
trainingF <- dfF[sets,]
testingF <- dfF[-sets,]
```

Análisis exploratorio de los datos

Realizo un analisis exploratorio en ambos dataframes utilizando análisis de componentes principales. Podemos observar que minutos y precio estan correlacionados, tiene sentido pensar que mientras menos tiempo tarde en llegar el estudiante a la Universidad Sapienza mas caro es la habitación. El dataset son apartamentos que ofrece la Sapienza para estudiantes. Sin embargo no sabemos mediante cuales medios de transporte pudieron realizar estos precios, es posible que si el apartamento queda a 20 minutos a pie de la universidad le hayan dado cierto valor, sin embargo es posible que un apartamento quede a 20 minutos en metro.

```
#*********************************************************************************
#----------------------ANALISIS EXPLORATORIO DE LOS DATOS-----------------------------
#*********************************************************************************
#---------------EN TEORIA EN OTRA PC ESTO DEBE CORRER-------------
#Selecciono las variables que puedo utilizar para aplicar regresion lineal
#para realizar un analisis exploratorio de los datos
#Femenino <- select(dfF, Minutos, TodoIncluido, TipoHabitacion, Pasillo, Cocina, Cuarto, Bagno, #Balcon

#PCA <- PCA(Femenino)
#plot(PCA, choix = "var")

#Masculino <- select(dfM, Minutos, TodoIncluido, TipoHabitacion, Pasillo, Cocina,Cuarto, Bagno, #Balcon

#PCA <- PCA(Masculino)
#plot(PCA, choix = "var")
```

Habitación ideal mediante la generación de pesos o valor

Obtengo la habitación ideal para un estudiante masculino que esta dispuesto a pagar precios estandares y economicos, utilizando el valor total y el precio.

```
#OBTENGO LA MEJOR HABITACION PARA HOMBRE.
dfM <- dfM[order(dfM$PrecioTotal) , ]
medianaMM <- median(dfM$PrecioTotal)
mejoresHabitacionesM <- subset(dfM , PrecioTotal < medianaMM)
mejorHabitacionesM <- mejoresHabitacionesM[which(mejoresHabitacionesM$ValorTotal == max(mejoresHabitaci
print(mejorHabitacionesM$Distrito)
```

```
## [1] "Bologna"
```

```
print(mejorHabitacionesM$Dirección)
```

```
## [1] "Via Gatteschi"
```

```
print(mejorHabitacionesM$Descripción)
```

```
## [1] "Ingresso/soggiorno, 2 camere, cucina, 2 bagni"
```

```
print(mejorHabitacionesM$PrecioTotal)
```

```
## [1] 400
```

```
print(mejorHabitacionesM$ValorTotal)
```

```
## [1] 9.059561
```

Obtengo la habitación ideal para un estudiante femenino que esta dispuesto a pagar precios estandares y economicos, utilizando el valor total y el precio.

```
#OBTENGO LA MEJOR HABITACION PARA LAS MUJERES.
dfF <- dfF[order(dfF$PrecioTotal) , ]
medianaMF <- median(dfF$PrecioTotal)
mejoresHabitacionesF <- subset(dfF , PrecioTotal < medianaMF)
mejorHabitacionesF <- mejoresHabitacionesF[which(mejoresHabitacionesF$ValorTotal == max(mejoresHabitaci
print(mejorHabitacionesF$Distrito)
```

```
## [1] "Bologna"
```

```
print(mejorHabitacionesF$Dirección)
```

```
## [1] "Via Gatteschi"
```

```r
print(mejorHabitacionesF$Descripción)
```

```
## [1] "Ingresso/soggiorno, 2 camere, cucina, 2 bagni"
```

```r
print(mejorHabitacionesF$PrecioTotal)
```

```
## [1] 400
```

```r
print(mejorHabitacionesF$ValorTotal)
```

```
## [1] 9.059561
```

Selección de variables para regresión lineal

Gracias al análisis exploratorio de los datos pude observar que no hay una variable que tenga una relación fuerte con el precio, es decir un X que me prediga un Y (regresión simple). Pero hay varias variables que si tienen un nivel de correlación con el precio, por lo tanto una regresión lineal múltiple me parece adecuado para obtener una predicción del precio más precisa, sin embargo elegir cuales variables es algo que hay que analizar. Utilizando PCA tuve una idea aproximada, pero utilizaré la selección de variables por pasos (forward, backward, both) utilizando la función stepAIC () del paquete MASS. stepAIC () realiza la selección del modelo paso a paso por la AIC exacta.

Primero aplico regresión lineal con todas las variables con el dataset de sexo masculino.

```r
#-----------------------------MASCULINO-----------------------------------------
#Aplico regresion lineal con todas las variables
modeloM1 <- lm(PrecioTotal ~ Minutos + TodoIncluido + TipoHabitacion + Pasillo +
               Cocina + Cuarto + Bagno + Balcon + Comedor + Armario + Salon +
               ValorTotal,data = trainingM)

regresionM1 <- predict(modeloM1, newdata = testingM)
```

```
## Warning in predict.lm(modeloM1, newdata = testingM): prediction from a
## rank-deficient fit may be misleading
```

```r
#-----------------------------FIN MASCULINO-------------------------------------
```

Evalúo con la función stepAIC, y busco las mejores variables para predecir.

```r
#Evaluo, y busco las mejores variables para predecir
step <- stepAIC(modeloM1, direction="both")
```

```
## Start:  AIC=622.77
## PrecioTotal ~ Minutos + TodoIncluido + TipoHabitacion + Pasillo +
##     Cocina + Cuarto + Bagno + Balcon + Comedor + Armario + Salon +
##     ValorTotal
##
##
## Step:  AIC=622.77
## PrecioTotal ~ Minutos + TodoIncluido + TipoHabitacion + Pasillo +
```

```
##     Cocina + Cuarto + Bagno + Balcon + Comedor + Armario + ValorTotal
##
##                   Df Sum of Sq    RSS    AIC
## - ValorTotal       1         7 453598 620.77
## - Comedor          1       200 453791 620.80
## - Armario          1       741 454332 620.88
## - Cocina           1      2181 455772 621.10
## - Bagno            1      7277 460868 621.85
## - Pasillo          1      7659 461250 621.91
## - Minutos          1      7824 461415 621.93
## - TodoIncluido     1      9997 463588 622.25
## - Balcon           1     11436 465027 622.46
## - Cuarto           1     13040 466631 622.70
## <none>                         453591 622.77
## - TipoHabitacion   1    467999 921590 668.98
##
## Step:  AIC=620.77
## PrecioTotal ~ Minutos + TodoIncluido + TipoHabitacion + Pasillo +
##     Cocina + Cuarto + Bagno + Balcon + Comedor + Armario
##
##                   Df Sum of Sq    RSS    AIC
## - Comedor          1       194 453792 618.80
## - Armario          1       742 454341 618.88
## - Cocina           1      2193 455791 619.10
## - Pasillo          1      7652 461250 619.91
## - Bagno            1      9934 463533 620.24
## - Balcon           1     11770 465368 620.51
## - Cuarto           1     13066 466665 620.70
## <none>                         453598 620.77
## + ValorTotal       1         7 453591 622.77
## - Minutos          1     29446 483045 623.05
## - TodoIncluido     1     37459 491058 624.17
## - TipoHabitacion   1    475382 928980 667.52
##
## Step:  AIC=618.8
## PrecioTotal ~ Minutos + TodoIncluido + TipoHabitacion + Pasillo +
##     Cocina + Cuarto + Bagno + Balcon + Armario
##
##                   Df Sum of Sq    RSS    AIC
## - Armario          1       690 454482 616.90
## - Cocina           1      2086 455878 617.11
## - Bagno            1      9835 463628 618.26
## - Pasillo          1     10018 463810 618.29
## - Balcon           1     12391 466183 618.63
## <none>                         453792 618.80
## - Cuarto           1     13902 467694 618.85
## + Comedor          1       194 453598 620.77
## + ValorTotal       1         1 453791 620.80
## - Minutos          1     29262 483054 621.05
## - TodoIncluido     1     37955 491747 622.26
## - TipoHabitacion   1    482334 936126 666.04
##
## Step:  AIC=616.9
## PrecioTotal ~ Minutos + TodoIncluido + TipoHabitacion + Pasillo +
```

```
##     Cocina + Cuarto + Bagno + Balcon
##
##                   Df Sum of Sq     RSS    AIC
## - Cocina           1      2132  456613 615.22
## - Bagno            1      9562  464044 616.32
## - Pasillo          1     10144  464626 616.40
## - Balcon           1     11862  466344 616.66
## <none>                          454482 616.90
## - Cuarto           1     14420  468902 617.03
## + Armario          1       690  453792 618.80
## + Comedor          1       141  454341 618.88
## + ValorTotal       1         0  454481 618.90
## - Minutos          1     28644  483126 619.06
## - TodoIncluido     1     38278  492760 620.40
## - TipoHabitacion   1    549048 1003530 668.77
##
## Step:  AIC=615.22
## PrecioTotal ~ Minutos + TodoIncluido + TipoHabitacion + Pasillo +
##     Cuarto + Bagno + Balcon
##
##                   Df Sum of Sq     RSS    AIC
## - Bagno            1     10284  466897 614.74
## - Pasillo          1     10929  467542 614.83
## - Balcon           1     12198  468811 615.01
## <none>                          456613 615.22
## + Cocina           1      2132  454482 616.90
## - Cuarto           1     26457  483070 617.05
## - Minutos          1     26513  483126 617.06
## + Armario          1       735  455878 617.11
## + Comedor          1        52  456562 617.21
## + ValorTotal       1        35  456578 617.22
## - TodoIncluido     1     37079  493692 618.53
## - TipoHabitacion   1    551772 1008386 667.10
##
## Step:  AIC=614.74
## PrecioTotal ~ Minutos + TodoIncluido + TipoHabitacion + Pasillo +
##     Cuarto + Balcon
##
##                   Df Sum of Sq     RSS    AIC
## - Pasillo          1     11346  478243 614.37
## - Balcon           1     12063  478960 614.47
## <none>                          466897 614.74
## - Cuarto           1     16591  483488 615.11
## + Bagno            1     10284  456613 615.22
## - Minutos          1     19839  486735 615.57
## + Cocina           1      2853  464044 616.32
## + ValorTotal       1      2313  464584 616.40
## + Armario          1       448  466449 616.67
## + Comedor          1       288  466609 616.69
## - TodoIncluido     1     38263  505160 618.09
## - TipoHabitacion   1    595715 1062612 668.66
##
## Step:  AIC=614.37
## PrecioTotal ~ Minutos + TodoIncluido + TipoHabitacion + Cuarto +
```

```
##     Balcon
##
##                   Df Sum of Sq     RSS    AIC
## - Cuarto           1     10733  488976 613.88
## <none>                         478243 614.37
## + Pasillo          1     11346  466897 614.74
## + Bagno            1     10700  467542 614.83
## - Minutos          1     19845  498087 615.13
## - Balcon           1     21130  499373 615.31
## + Comedor          1      6140  472102 615.49
## + ValorTotal       1      3921  474321 615.81
## + Cocina           1      3783  474460 615.83
## + Armario          1       558  477685 616.29
## - TodoIncluido     1     31531  509773 616.71
## - TipoHabitacion   1    591403 1069646 667.11
##
## Step:  AIC=613.88
## PrecioTotal ~ Minutos + TodoIncluido + TipoHabitacion + Balcon
##
##                   Df Sum of Sq     RSS    AIC
## <none>                         488976 613.88
## + Cocina           1     11235  477741 614.30
## + Cuarto           1     10733  478243 614.37
## - Minutos          1     22044  511020 614.88
## - Balcon           1     22569  511544 614.95
## + Pasillo          1      5488  483488 615.11
## + Armario          1      1433  487543 615.68
## + Comedor          1      1171  487805 615.72
## + Bagno            1      1034  487942 615.73
## + ValorTotal       1       215  488761 615.85
## - TodoIncluido     1     32045  521021 616.19
## - TipoHabitacion   1    924015 1412991 684.04
```

```
step$anova #muestro los resultados
```

```
## Stepwise Model Path
## Analysis of Deviance Table
##
## Initial Model:
## PrecioTotal ~ Minutos + TodoIncluido + TipoHabitacion + Pasillo +
##     Cocina + Cuarto + Bagno + Balcon + Comedor + Armario + Salon +
##     ValorTotal
##
## Final Model:
## PrecioTotal ~ Minutos + TodoIncluido + TipoHabitacion + Balcon
##
##
##           Step Df    Deviance Resid. Df Resid. Dev      AIC
## 1                                    56   453591.0 622.7702
## 2      - Salon  0     0.00000        56   453591.0 622.7702
## 3 - ValorTotal  1     7.45669        57   453598.4 620.7713
## 4    - Comedor  1   193.64502        58   453792.1 618.8003
## 5    - Armario  1   689.52688        59   454481.6 616.9035
## 6     - Cocina  1  2131.58435        60   456613.2 615.2217
```

```
## 7      - Bagno  1 10283.72443        61   466896.9 614.7362
## 8    - Pasillo  1 11345.67093        62   478242.6 614.3689
## 9     - Cuarto  1 10733.19993        63   488975.8 613.8781
```

Aplico regresión lineal con las variables seleccionadas por la función stepAIC.

```
modeloM2 <- lm(PrecioTotal ~ Minutos + TodoIncluido + TipoHabitacion + Balcon,data = trainingM)

regresionM2 <- predict(modeloM2, newdata = testingM)
```

Comparación entre ambos modelos de regresión lineal

```
anova(modeloM1, modeloM2)
```

```
## Analysis of Variance Table
##
## Model 1: PrecioTotal ~ Minutos + TodoIncluido + TipoHabitacion + Pasillo +
##     Cocina + Cuarto + Bagno + Balcon + Comedor + Armario + Salon +
##     ValorTotal
## Model 2: PrecioTotal ~ Minutos + TodoIncluido + TipoHabitacion + Balcon
##   Res.Df    RSS Df Sum of Sq      F Pr(>F)
## 1     56 453591
## 2     63 488976 -7    -35385 0.6241 0.7338
```

```
comparacionM <- cbind(regresionM2,testingM$PrecioTotal )
```

Primero aplico regresión lineal con todas las variables con el dataset de sexo femenino.

```
#-----------------------------FEMENINO-----------------------------------------
#Aplico regresion lineal con todas las variables
modeloF1 <- lm(PrecioTotal ~ Minutos + TodoIncluido + TipoHabitacion + Pasillo +
                Cocina + Cuarto + Bagno + Balcon + Comedor + Armario + Salon +
                ValorTotal,data = trainingF)

regresionF1 <- predict(modeloF1, newdata = testingF)
```

Evalúo con la función stepAIC, y busco las mejores variables para predecir.

```
step <- stepAIC(modeloF1, direction="both")
```

```
## Start:  AIC=919.08
## PrecioTotal ~ Minutos + TodoIncluido + TipoHabitacion + Pasillo +
##     Cocina + Cuarto + Bagno + Balcon + Comedor + Armario + Salon +
##     ValorTotal
##
##                  Df Sum of Sq     RSS    AIC
## - Salon           1        34  600474 917.09
## - Bagno           1        97  600537 917.10
## - Cuarto          1       779  601219 917.21
```

```
## - ValorTotal      1       847  601288 917.23
## - Cocina          1      1499  601939 917.34
## - Minutos         1      2419  602859 917.50
## - Armario         1      5121  605561 917.96
## - Comedor         1      7462  607903 918.35
## <none>                         600440 919.08
## - Pasillo         1     11827  612268 919.09
## - TodoIncluido    1     18836  619276 920.26
## - Balcon          1     26427  626867 921.52
## - TipoHabitacion  1    564586 1165026 985.35
##
## Step:  AIC=917.09
## PrecioTotal ~ Minutos + TodoIncluido + TipoHabitacion + Pasillo +
##     Cocina + Cuarto + Bagno + Balcon + Comedor + Armario + ValorTotal
##
##                  Df Sum of Sq     RSS    AIC
## - Bagno           1       135  600609 915.11
## - ValorTotal      1       893  601367 915.24
## - Cuarto          1       952  601426 915.25
## - Cocina          1      1468  601942 915.34
## - Minutos         1      2393  602867 915.50
## - Armario         1      5126  605600 915.96
## - Comedor         1      7490  607964 916.36
## <none>                         600474 917.09
## - Pasillo         1     11800  612274 917.09
## - TodoIncluido    1     19001  619475 918.30
## + Salon           1        34  600440 919.08
## - Balcon          1     26515  626988 919.54
## - TipoHabitacion  1    572132 1172606 984.02
##
## Step:  AIC=915.11
## PrecioTotal ~ Minutos + TodoIncluido + TipoHabitacion + Pasillo +
##     Cocina + Cuarto + Balcon + Comedor + Armario + ValorTotal
##
##                  Df Sum of Sq     RSS    AIC
## - ValorTotal      1       771  601380 913.24
## - Cuarto          1       819  601427 913.25
## - Cocina          1      1763  602372 913.41
## - Minutos         1      3245  603854 913.66
## - Armario         1      5139  605747 913.99
## - Comedor         1      7384  607992 914.37
## <none>                         600609 915.11
## - Pasillo         1     12385  612994 915.21
## - TodoIncluido    1     21557  622165 916.74
## + Bagno           1       135  600474 917.09
## + Salon           1        72  600537 917.10
## - Balcon          1     26906  627515 917.62
## - TipoHabitacion  1    593861 1194469 983.92
##
## Step:  AIC=913.24
## PrecioTotal ~ Minutos + TodoIncluido + TipoHabitacion + Pasillo +
##     Cocina + Cuarto + Balcon + Comedor + Armario
##
##                  Df Sum of Sq     RSS    AIC
```

```
## - Cuarto          1      1287  602667 911.46
## - Cocina          1      2711  604091 911.71
## - Armario         1      5297  606677 912.15
## - Comedor         1      6841  608221 912.41
## <none>                         601380 913.24
## - Pasillo         1     12759  614139 913.40
## - Minutos         1     20655  622035 914.72
## + ValorTotal      1       771  600609 915.11
## + Salon           1        62  601318 915.23
## + Bagno           1        13  601367 915.24
## - Balcon          1     27314  628694 915.82
## - TodoIncluido    1     53089  654469 919.96
## - TipoHabitacion  1    599329 1200709 982.46
##
## Step:  AIC=911.46
## PrecioTotal ~ Minutos + TodoIncluido + TipoHabitacion + Pasillo +
##     Cocina + Balcon + Comedor + Armario
##
##                   Df Sum of Sq     RSS    AIC
## - Cocina           1      5005  607672 910.31
## - Armario          1      5764  608432 910.44
## - Comedor          1      6967  609634 910.65
## <none>                         602667 911.46
## - Pasillo          1     14012  616680 911.83
## + Cuarto           1      1287  601380 913.24
## + ValorTotal       1      1240  601427 913.25
## + Bagno            1       395  602272 913.39
## + Salon            1       224  602444 913.42
## - Balcon           1     27020  629687 913.98
## - Minutos          1     28741  631408 914.26
## - TodoIncluido     1     52464  655132 918.06
## - TipoHabitacion   1    671063 1273730 986.54
##
## Step:  AIC=910.31
## PrecioTotal ~ Minutos + TodoIncluido + TipoHabitacion + Pasillo +
##     Balcon + Comedor + Armario
##
##                   Df Sum of Sq     RSS    AIC
## - Armario          1      6122  613795 909.35
## - Comedor          1      6251  613923 909.37
## <none>                         607672 910.31
## - Pasillo          1     13589  621261 910.59
## + Cocina           1      5005  602667 911.46
## + Cuarto           1      3581  604091 911.71
## + ValorTotal       1      3504  604168 911.72
## + Bagno            1       832  606840 912.17
## + Salon            1       276  607397 912.27
## - Minutos          1     24366  632038 912.36
## - Balcon           1     26651  634323 912.74
## - TodoIncluido     1     52084  659757 916.78
## - TipoHabitacion   1    696090 1303763 986.94
##
## Step:  AIC=909.35
## PrecioTotal ~ Minutos + TodoIncluido + TipoHabitacion + Pasillo +
```

```
##     Balcon + Comedor
##
##                   Df Sum of Sq     RSS    AIC
## - Comedor          1      5442  619236 908.26
## <none>                         613795 909.35
## - Pasillo          1     12173  625968 909.37
## + Armario          1      6122  607672 910.31
## + Cocina           1      5363  608432 910.44
## + Cuarto           1      4411  609383 910.60
## + ValorTotal       1      4098  609697 910.66
## - Minutos          1     21299  635094 910.86
## - Balcon           1     22618  636413 911.07
## + Bagno            1      1082  612713 911.17
## + Salon            1       340  613454 911.29
## - TodoIncluido     1     53583  667378 915.97
## - TipoHabitacion   1    858229 1472023 997.44
##
## Step:  AIC=908.26
## PrecioTotal ~ Minutos + TodoIncluido + TipoHabitacion + Pasillo +
##     Balcon
##
##                   Df Sum of Sq     RSS    AIC
## - Pasillo          1      6882  626118 907.39
## <none>                         619236 908.26
## - Minutos          1     17515  636751 909.13
## - Balcon           1     18417  637653 909.27
## + Comedor          1      5442  613795 909.35
## + Armario          1      5313  613923 909.37
## + Cocina           1      4642  614595 909.48
## + Cuarto           1      4302  614935 909.54
## + ValorTotal       1      2649  616587 909.81
## + Bagno            1      1026  618211 910.09
## + Salon            1       366  618870 910.19
## - TodoIncluido     1     52055  671291 914.57
## - TipoHabitacion   1    852877 1472113 995.45
##
## Step:  AIC=907.39
## PrecioTotal ~ Minutos + TodoIncluido + TipoHabitacion + Balcon
##
##                   Df Sum of Sq     RSS    AIC
## <none>                         626118 907.39
## - Balcon           1     14163  640282 907.70
## + Pasillo          1      6882  619236 908.26
## - Minutos          1     18443  644561 908.38
## + Cuarto           1      6079  620039 908.39
## + ValorTotal       1      5053  621066 908.56
## + Cocina           1      4997  621121 908.57
## + Armario          1      4723  621395 908.61
## + Bagno            1       839  625279 909.26
## + Salon            1       465  625654 909.32
## + Comedor          1       151  625968 909.37
## - TodoIncluido     1     58314  684433 914.57
## - TipoHabitacion   1    852496 1478614 993.90
```

```
step$anova # display results
```

```
## Stepwise Model Path
## Analysis of Deviance Table
##
## Initial Model:
## PrecioTotal ~ Minutos + TodoIncluido + TipoHabitacion + Pasillo +
##     Cocina + Cuarto + Bagno + Balcon + Comedor + Armario + Salon +
##     ValorTotal
##
## Final Model:
## PrecioTotal ~ Minutos + TodoIncluido + TipoHabitacion + Balcon
##
##
##           Step Df   Deviance Resid. Df Resid. Dev      AIC
## 1                                   90   600440.1 919.0810
## 2      - Salon  1   33.56488        91   600473.7 917.0867
## 3      - Bagno  1  135.09283        92   600608.8 915.1099
## 4 - ValorTotal  1  771.43288        93   601380.2 913.2421
## 5     - Cuarto  1 1287.28569        94   602667.5 911.4624
## 6     - Cocina  1 5004.86261        95   607672.3 910.3142
## 7    - Armario  1 6122.35916        96   613794.7 909.3467
## 8    - Comedor  1 5441.54549        97   619236.3 908.2559
## 9    - Pasillo  1 6881.94096        98   626118.2 907.3942
```

Aplico regresión lineal con las variables seleccionadas por la función stepAIC.

```
modeloF2 <- lm(PrecioTotal ~ Minutos + TodoIncluido + TipoHabitacion + Balcon,data = trainingF)

regresionF2 <- predict(modeloF2, newdata = testingF)
```

Comparación entre ambos modelos de regresión lineal

```
anova(modeloM1, modeloM2)
```

```
## Analysis of Variance Table
##
## Model 1: PrecioTotal ~ Minutos + TodoIncluido + TipoHabitacion + Pasillo +
##     Cocina + Cuarto + Bagno + Balcon + Comedor + Armario + Salon +
##     ValorTotal
## Model 2: PrecioTotal ~ Minutos + TodoIncluido + TipoHabitacion + Balcon
##   Res.Df    RSS Df Sum of Sq      F Pr(>F)
## 1     56 453591
## 2     63 488976 -7    -35385 0.6241 0.7338
```

```
comparacionF <- cbind(regresionF2,testingF$PrecioTotal )
```

Habitación ideal para el estudiante utilizando regresión lineal

Se puede decidir cual es el inmueble adecuado para el estudiante utilizando regresión lineal, se toman los apartamentos con precios intermedios y bajos, y luego seleccionamos el precio más alto que propuso la regresión lineal.

```
#utilizando regresion lineal para decidir el mejor apto.
#OBTENGO LA MEJOR HABITACION PARA HOMBRE.
hombre <- lm(PrecioTotal ~ Minutos + TodoIncluido + TipoHabitacion + Balcon,data = dfM)

hombre <- predict(hombre, newdata = dfM)
dfM$PrediccionPrecio <- cbind(hombre)

dfM <- dfM[order(dfM$PrecioTotal) , ]
medianaMM <- median(dfM$PrecioTotal)
mejoresHabitacionesM <- subset(dfM , PrecioTotal < medianaMM)
mejorHabitacionesM <- mejoresHabitacionesM[which(mejoresHabitacionesM$PrediccionPrecio == max(mejoresHa
print(mejorHabitacionesM$Distrito)
```

```
## [1] "Centro\n/ Manzoni"
```

```
print(mejorHabitacionesM$Dirección)
```

```
## [1] "via Alfieri int. 12"
```

```
print(mejorHabitacionesM$Descripción)
```

```
## [1] "Ingresso, soggiorno, 3 camere, cucina/living, bagno"
```

```
print(mejorHabitacionesM$PrecioTotal)
```

```
## [1] 475
```

```
print(mejorHabitacionesM$ValorTotal)
```

```
## [1] 4.043887
```

```
#OBTENGO LA MEJOR HABITACION PARA LAS MUJERES.
mujer <- lm(PrecioTotal ~ Minutos + TodoIncluido + TipoHabitacion + Balcon,data = dfF)

mujer <- predict(mujer, newdata = dfF)
dfF$PrediccionPrecio <- cbind(mujer)

dfF <- dfF[order(dfF$PrecioTotal) , ]
medianaMF <- median(dfF$PrecioTotal)
mejoresHabitacionesF <- subset(dfF , PrecioTotal < medianaMF)
mejorHabitacionesF <- mejoresHabitacionesF[which(mejoresHabitacionesF$PrediccionPrecio == max(mejoresHa
print(mejorHabitacionesF$Distrito)
```

```
## [1] "Centocelle"
```

```r
print(mejorHabitacionesF$Dirección)
```

```
## [1] "Via delle Palme"
```

```r
print(mejorHabitacionesF$Descripción)
```

```
## [1] "Ingresso, cucina, 2 camere, bagno"
```

```r
print(mejorHabitacionesF$PrecioTotal)
```

```
## [1] 350
```

```r
print(mejorHabitacionesF$ValorTotal)
```

```
## [1] 3.573668
```