Buscando un Hogar

Deyban Pérez

March 11, 2016

## Abstract

El amigo de un amigo fue admitido en la **Universidad de Sapienza** de Roma, este amigo vive en Venezuela y necesita elegir un hogar de manera inteligente ya que en nuestro país el contro cambiario nos tiene a todos la vida triste y necesita ahorrar la mayor cantidad de dinero posible. Para ello me pidió el favor de que lo ayduara a estimar el valor d eun buen apartamento, situación que lleva a la elaboración de la presente asignación: **Estimar el costo de una viviena basandose en sus características**, adicionalmente se debe elegir una vivienda para el amigo de mi amigo y crear un modelo tanto para hombres como para mujeres. La problemática se solucionó utilizando un modelo de **regresión lineal** para estimar los costos y una función de pesos para calcular el hogar del amigo de mi amigo.

## Actividades Realizadas

### Preprocesamiento

Se realizó un preprocesamiento sobre el **dataset** debido a que las columnas inicialmente no aportaban información extraible poruqe no estaban categorizadas ni estandarizadas. OPor lo mismo el primer paso fue crear una vista minable sobre la cual se pudiera trabajar

### Agregando Información de Distancia Hacia la Universidad

Cómo se mencionón anteriormenet, el amigo de un amigo bsuca ir a Roma a estudiar, y le caería perfecto que su nuevo hogare stuviera cerca de la universidad, por lo mismo, se utilizó el **API de Google** para calcular el tiempo necesraio en carro para poder llegar desde cada uno de los hogares disponibles hasta la universidad.

A continuación se muestran los priemros pasos que corresponden a los pasos de eliminación de algunas columnas nulas y el arreglo d ela información para solicitar la información al API de Google.

#begin  
#######################################################################  
#Funcion para normalizar una columna de un data frame  
normalize = function(x)  
{  
 num <- x - min(x)  
 denom <- max(x) - min(x)  
 return (num/denom)  
}  
#################################################################################  
# Seleccionar google\_api.R en su sistema de archivos  
source("google\_api.R")

## Loading required package: jsonlite

#################################################################################  
# Installing required packages  
install("curl")

## Loading required package: curl

install("xlsx")

## Loading required package: xlsx

## Loading required package: rJava

## Loading required package: xlsxjars

#################################################################################  
#loading required libraries  
library("xlsx")  
#################################################################################  
#Loading dataset  
mydata = read.xlsx("hogares.xlsx", sheetIndex = 1, startRow = 1, endRow = 104, header = T)  
#################################################################################  
#Preprocessing Dataset  
#################################################################################  
#Deleting irrelevant column  
mydata$Foto = NULL  
mydata$Piso = NULL  
#################################################################################  
#Removing \n in Column Direction  
mydata$Dirección = strsplit(as.character(mydata$Dirección), "\n")  
  
for(i in seq(1:nrow(mydata)))  
{  
 mydata$Dirección[i] = paste(mydata$Dirección[[i]], collapse = " ")  
   
}  
##################################################################################  
#Converting Direccion Column into factor type  
mydata$Dirección = as.character(mydata$Dirección)  
##################################################################################  
#Fixing typos in som Directions  
mydata$Dirección[11] = "Galliate"  
mydata$Dirección[33] = "Via San Roberto Bellarmino"  
mydata$Dirección[61] = "Via di Monte Verde"  
##################################################################################  
#Adding Distance column to dataset  
mydata$Time = 0  
##################################################################################  
#setting goal position  
destino =c("Piazzale Aldo Moro")  
##################################################################################  
#Setting API key  
api\_key = "AIzaSyCJ-6caC04NXNHDjpLq0BV2C9obZINR6bo"  
#################################################################################  
#Cicle to fill time column  
for(i in seq(1:nrow(mydata)))  
{  
 #Setting start position  
 origen = mydata$Dirección[i]  
 #Making the URL  
 api\_url = get\_url(origen, destino, api\_key)  
 #Getting the data from API  
 datos = get\_data(api\_url)  
 #PArsing Json to Dataframe  
 retorno = parse\_data(datos)  
 #If the response from API is not NULL  
 if(retorno$status == "OK")  
 {  
 #Making split to transform hours to mins  
 aux1 = strsplit(as.character(retorno$duration$text), " ")  
 aux1 = paste(aux1[[1]], collapse = "")  
 aux1 = strsplit(as.character(aux1), "min")  
 aux1 = strsplit(as.character(aux1), "h")  
 #If is only has minutes  
 if (length(aux1[[1]]) == 1)  
 {  
 mydata$Time[i] = aux1[[1]][1]  
 }  
 else #If has hours  
 {  
 mydata$Time[i] = (as.integer(aux1[[1]][1])\*60) + as.integer(aux1[[1]][2])  
   
 }  
 }  
}

### Numerizando las Columnas Presentes

Se numerizaron las mayor cantidad de columnas categóricas posibles, emepzando por extraer por extraer palabras clave de la columna **Notas** sobre el sexo de la persona y categorizandolo como 0 (Hombres), 1 (Mujerez), 2 (Ambos sexos) y haciendo lo mismo para el **tipo de vivienda** 0 (Apartamento), 1 (Miniapartamento), 2 (Estudio).

####################################################################################################################  
#Numerizing column Notas, 0 for boys, 1 for girls, 2 for both  
mydata$Notas = as.character(mydata$Notas)  
mydata$Notas[grepl("(ragazze/i) | (ragazzi/e) | (ragazzi/ragazze) | (ragazze/ragazzi)", mydata$Notas)] = as.character(2)  
mydata$Notas[grepl("(ragazzi)", mydata$Notas)] = as.character(0)  
mydata$Notas[grepl("(ragazze)", mydata$Notas)] = as.character(1)  
mydata$Notas[39] = as.character(2)  
mydata$Notas = as.factor(mydata$Notas)  
#####################################################################################################################  
#Numerizing column Tipo.de.Inmueble, 0 fo Apartments, 1 for mini Apartments, 2 for Monolocale  
levels(mydata$Tipo.de.Inmueble)

## [1] "Apartamento" "Apparrtamento" "Appartameno"   
## [4] "Appartamenti" "Appartamento" "Mini appartamento"   
## [7] "Mini\nAppartamento" "Monolocale"

mydata$Tipo.de.Inmueble = as.character(mydata$Tipo.de.Inmueble)  
mydata$Tipo.de.Inmueble[mydata$Tipo.de.Inmueble == "Apartamento"] = as.character(0)  
mydata$Tipo.de.Inmueble[mydata$Tipo.de.Inmueble == "Apparrtamento"] = as.character(0)  
mydata$Tipo.de.Inmueble[mydata$Tipo.de.Inmueble == "Appartameno"] = as.character(0)  
mydata$Tipo.de.Inmueble[mydata$Tipo.de.Inmueble == "Appartamenti"] = as.character(0)  
mydata$Tipo.de.Inmueble[mydata$Tipo.de.Inmueble == "Appartamento"] = as.character(0)  
mydata$Tipo.de.Inmueble[mydata$Tipo.de.Inmueble == "Mini appartamento"] = as.character(1)  
mydata$Tipo.de.Inmueble[mydata$Tipo.de.Inmueble == "Mini\nAppartamento"] = as.character(1)  
mydata$Tipo.de.Inmueble[mydata$Tipo.de.Inmueble == "Monolocale"] = as.character(2)  
mydata$Tipo.de.Inmueble = as.factor(mydata$Tipo.de.Inmueble)  
######################################################################################################################

### Agregando Otras características Importantes ue se Pueden Extraer de Otras Columnas

De las oclumnas existentes de se pudo extraer características asociadas a la vivienda que no estaban separadasy se separarron y se agregaron esas nuevas características al dataset

######################################################################################################################  
#Creating otehr feature to avaluate  
mydata$Ingresso = 0  
mydata$Soggiorno = 0  
mydata$Cucina = 0  
mydata$Bagno = 0  
mydata$Salone = 0  
mydata$Disimpegno = 0  
mydata$Corridoio = 0  
mydata$Internet = 0  
mydata$Ripostiglio = 0  
mydata$Balcone = 0  
mydata$Termo = 0  
mydata$Terrazzo = 0  
#######################################################################################################################  
#Filling new features with description   
#######################################################################################################################  
#Filling Ingresso column (entrada)  
mydata$Ingresso[grepl("(Ingresso)", mydata$Descripción)] = 1  
mydata$Ingresso[grepl("(ingresso)", mydata$Descripción)] = 1  
#######################################################################################################################  
#Filling Soggiorno column (Sala de estar)  
mydata$Soggiorno[grepl("(soggiorno)", mydata$Descripción)] = 1  
#######################################################################################################################  
#Filling Cucina column (Cocina)  
mydata$Cucina[grepl("(cucina)", mydata$Descripción)] = 1  
mydata$Cucina[grepl("(cucna)", mydata$Descripción)] = 1  
mydata$Cucina[grepl("(cottura)", mydata$Descripción)] = 1  
#######################################################################################################################  
#Filling Bagno column (Bano)  
#######################################################################################################################  
# For 1 bagno  
mydata$Bagno[grepl("(bagno)", mydata$Descripción)] = 1  
mydata$Bagno[grepl("(1 bagno)", mydata$Descripción)] = 1  
#For 2 bagni  
mydata$Bagno[grepl("(2 bagni)", mydata$Descripción)] = 2  
mydata$Bagno[grepl("(2bagni)", mydata$Descripción)] = 2  
#For 3 bagni  
mydata$Bagno[grepl("(3 bagni)", mydata$Descripción)] = 3  
mydata$Bagno[grepl("(3bagni)", mydata$Descripción)] = 3  
#For 4 bagni  
mydata$Bagno[grepl("(4 bagni)", mydata$Descripción)] = 4  
mydata$Bagno[grepl("(4bagni)", mydata$Descripción)] = 4  
#######################################################################################################################  
#Filling Salone column (Sala de estar)  
mydata$Salone[grepl("(salone)", mydata$Descripción)] = 1  
mydata$Salone[grepl("(salotto)", mydata$Descripción)] = 1  
mydata$Salone[grepl("(sala da pranzo)", mydata$Descripción)] = 1  
#######################################################################################################################  
#Filling Disimpegno column (Salida de emergencia)  
mydata$Disimpegno[grepl("(disimpegno)", mydata$Descripción)] = 1  
#######################################################################################################################  
#Filling Corridoio column (Corredor)  
mydata$Corridoio[grepl("(corridoio)", mydata$Descripción)] = 1  
#######################################################################################################################  
#Filling internet column (internet)  
mydata$Internet[grepl("(internet)", mydata$Descripción)] = 1  
#######################################################################################################################  
#Filling ripostiglio column (Armario)  
mydata$Ripostiglio[grepl("(ripostiglio)", mydata$Descripción)] = 1  
#######################################################################################################################  
#Filling Balcone column (Balcon)  
mydata$Balcone[grepl("(balcone)", mydata$Descripción)] = 1  
mydata$Balcone[grepl("(balconcino)", mydata$Descripción)] = 1  
#######################################################################################################################  
#Filling termo column (calefaccion)  
mydata$Termo[grepl("(termo)", mydata$Descripción)] = 1  
#######################################################################################################################  
#Filling Terrazo column (terraza)  
mydata$Terrazzo[grepl("(terrazzo)", mydata$Descripción)] = 1  
#######################################################################################################################  
#Deleting columns that will not use again  
mydata$Descripción = NULL

### Separación de Individuos en Múltiples Filas

Los apartamentos podían tener diferentes precios teniendo las mismas características en cuyo caso se tomo el menor precio, o variar de precio si el aṕartamento era simple o compartido, en cuyo caso se duplicó la información y se agregó un nuevo individuo en la fila.

#######################################################################################################################  
#Converting types of features to a easiest data type (character)  
mydata$Habitaciones.Disponibles = as.character(mydata$Habitaciones.Disponibles)  
mydata$Precio.Mensual= as.character(mydata$Precio.Mensual)  
mydata$Distrito = as.character(mydata$Distrito)  
mydata$Dirección = as.character(mydata$Dirección)  
#######################################################################################################################  
#Splitting rows from columns with multiple disposition  
new\_row = c(as.character(mydata$Distrito[4]), as.character(mydata$Dirección[4]), as.character(mydata$Tipo.de.Inmueble[4]), "doppia", "€ 300 condominio, acqua, riscaldamento, internet inclusi", as.character(mydata$Notas[4]), as.character(mydata$Time[4]), as.character(mydata$Ingresso[4]), as.character(mydata$Soggiorno[4]), as.character(mydata$Cucina[4]), as.character(mydata$Bagno[4]), as.character(mydata$Salone[4]), as.character(mydata$Disimpegno[4]), as.character(mydata$Corridoio[4]), as.character(mydata$Internet[4]), as.character(mydata$Ripostiglio[4]), as.character(mydata$Balcone[4]), as.character(mydata$Termo[4]), as.character(mydata$Terrazzo[4]))  
mydata <- rbind(mydata, new\_row)  
mydata$Habitaciones.Disponibles[4] = "singola"  
mydata$Precio.Mensual[4] = "€ 450 condominio, acqua, riscaldamento, internet inclusi"  
########################################################################################################################  
mydata$Habitaciones.Disponibles[5] = "singola"  
mydata$Precio.Mensual[5] = "€ 505"  
########################################################################################################################  
new\_row = c(as.character(mydata$Distrito[7]), as.character(mydata$Dirección[7]), as.character(mydata$Tipo.de.Inmueble[7]), "doppia", "€ 450", as.character(mydata$Notas[7]), as.character(mydata$Time[7]), as.character(mydata$Ingresso[7]), as.character(mydata$Soggiorno[7]), as.character(mydata$Cucina[7]), as.character(mydata$Bagno[7]), as.character(mydata$Salone[7]), as.character(mydata$Disimpegno[7]), as.character(mydata$Corridoio[7]), as.character(mydata$Internet[7]), as.character(mydata$Ripostiglio[7]), as.character(mydata$Balcone[7]), as.character(mydata$Termo[7]), as.character(mydata$Terrazzo[7]))  
mydata <- rbind(mydata, new\_row)  
mydata$Habitaciones.Disponibles[7] = "singola"  
mydata$Precio.Mensual[7] = "€ 250"  
########################################################################################################################  
new\_row = c(as.character(mydata$Distrito[10]), as.character(mydata$Dirección[10]), as.character(mydata$Tipo.de.Inmueble[10]), "doppia", "€ 350 TUTTO INCLUSO", as.character(mydata$Notas[10]), as.character(mydata$Time[10]), as.character(mydata$Ingresso[10]), as.character(mydata$Soggiorno[10]), as.character(mydata$Cucina[10]), as.character(mydata$Bagno[10]), as.character(mydata$Salone[10]), as.character(mydata$Disimpegno[10]), as.character(mydata$Corridoio[10]), as.character(mydata$Internet[10]), as.character(mydata$Ripostiglio[10]), as.character(mydata$Balcone[10]), as.character(mydata$Termo[10]), as.character(mydata$Terrazzo[10]))  
mydata$Habitaciones.Disponibles[10] = "doppia"  
mydata <- rbind(mydata, new\_row)  
mydata$Habitaciones.Disponibles[10] = "singola"  
mydata$Precio.Mensual[10] = "€ 550 TUTTO INCLUSO"  
########################################################################################################################  
mydata$Habitaciones.Disponibles[16] = "singola"  
mydata$Precio.Mensual[16] = "€ 430"  
########################################################################################################################  
mydata$Habitaciones.Disponibles[18] = "singola"  
mydata$Precio.Mensual[18] = "€ 450"  
########################################################################################################################  
mydata$Habitaciones.Disponibles[21] = "singola"  
mydata$Precio.Mensual[21] = "€ 475 TUTTO INCLUSO"  
########################################################################################################################  
mydata$Habitaciones.Disponibles[22] = "singola"  
mydata$Precio.Mensual[22] = "€ 525 TUTTO INCLUSO"  
#######################################################################################################################  
mydata$Habitaciones.Disponibles[23] = "singola"  
mydata$Precio.Mensual[23] = "€ 475 TUTTO INCLUSO"  
#######################################################################################################################  
mydata$Habitaciones.Disponibles[24] = "singola"  
mydata$Precio.Mensual[24] = "€ 475 internet incluso"  
#######################################################################################################################  
mydata$Habitaciones.Disponibles[26] = "singola"  
mydata$Precio.Mensual[26] = "€ 380 condominio e acqua inclusi"  
#######################################################################################################################  
new\_row = c(as.character(mydata$Distrito[31]), as.character(mydata$Dirección[31]), as.character(mydata$Tipo.de.Inmueble[31]), "doppia", "€ 250 condominio e riscaldamento", as.character(mydata$Notas[31]), as.character(mydata$Time[31]), as.character(mydata$Ingresso[31]), as.character(mydata$Soggiorno[31]), as.character(mydata$Cucina[31]), as.character(mydata$Bagno[31]), as.character(mydata$Salone[31]), as.character(mydata$Disimpegno[31]), as.character(mydata$Corridoio[31]), as.character(mydata$Internet[31]), as.character(mydata$Ripostiglio[31]), as.character(mydata$Balcone[31]), as.character(mydata$Termo[31]), as.character(mydata$Terrazzo[31]))  
mydata <- rbind(mydata, new\_row)  
mydata$Habitaciones.Disponibles[31] = "singola"  
mydata$Precio.Mensual[31] = "€ 450 condominio e riscaldamento"  
#######################################################################################################################  
mydata$Habitaciones.Disponibles[41] = "singola"  
mydata$Precio.Mensual[41] = "€ 350 condominio e acqua inclusi"  
#######################################################################################################################  
mydata$Habitaciones.Disponibles[42] = "singola"  
mydata$Precio.Mensual[42] = "€ 420 condominio, acqua, riscaldamento e tassa rifiuti inclusi"  
#######################################################################################################################  
mydata$Habitaciones.Disponibles[43] = "singola"  
mydata$Precio.Mensual[43] = "€ 450 condominio, acqua e tassa rifiuti inclusi"  
#######################################################################################################################  
mydata$Habitaciones.Disponibles[44] = "singola"  
mydata$Precio.Mensual[44] = "€ 340 spese escluse"  
#######################################################################################################################  
mydata$Habitaciones.Disponibles[56] = "singola"  
mydata$Precio.Mensual[56] = "€ 450 condominio, acqua e riscaldamento"  
#######################################################################################################################  
mydata$Habitaciones.Disponibles[61] = "singola"  
mydata$Precio.Mensual[61] = "€ 425 condominio, acqua e riscaldamento"  
#######################################################################################################################  
new\_row = c(as.character(mydata$Distrito[72]), as.character(mydata$Dirección[72]), as.character(mydata$Tipo.de.Inmueble[72]), "doppia", "€ 350 TUTTO INCLUSO", as.character(mydata$Notas[72]), as.character(mydata$Time[72]), as.character(mydata$Ingresso[72]), as.character(mydata$Soggiorno[72]), as.character(mydata$Cucina[72]), as.character(mydata$Bagno[72]), as.character(mydata$Salone[72]), as.character(mydata$Disimpegno[72]), as.character(mydata$Corridoio[72]), as.character(mydata$Internet[72]), as.character(mydata$Ripostiglio[72]), as.character(mydata$Balcone[72]), as.character(mydata$Termo[72]), as.character(mydata$Terrazzo[72]))  
mydata <- rbind(mydata, new\_row)  
mydata$Habitaciones.Disponibles[72] = "singola"  
mydata$Precio.Mensual[72] = "€ 450 TUTTO INCLUSO"  
#######################################################################################################################  
mydata$Habitaciones.Disponibles[74] = "singola"  
mydata$Precio.Mensual[74] = "€ 450 Spese escluse"  
#######################################################################################################################  
new\_row = c(as.character(mydata$Distrito[76]), as.character(mydata$Dirección[76]), as.character(mydata$Tipo.de.Inmueble[76]), "singola", "€ 425 condominio, acqua e riscaldamento inclusi", as.character(mydata$Notas[76]), as.character(mydata$Time[76]), as.character(mydata$Ingresso[76]), as.character(mydata$Soggiorno[76]), as.character(mydata$Cucina[76]), as.character(mydata$Bagno[76]), as.character(mydata$Salone[76]), as.character(mydata$Disimpegno[76]), as.character(mydata$Corridoio[76]), as.character(mydata$Internet[76]), as.character(mydata$Ripostiglio[76]), as.character(mydata$Balcone[76]), as.character(mydata$Termo[76]), as.character(mydata$Terrazzo[76]))  
mydata <- rbind(mydata, new\_row)  
mydata$Habitaciones.Disponibles[76] = "intero appartamento"  
mydata$Precio.Mensual[76] = "€ 850 condominio, acqua e riscaldamento inclusi"  
#######################################################################################################################  
new\_row = c(as.character(mydata$Distrito[77]), as.character(mydata$Dirección[77]), as.character(mydata$Tipo.de.Inmueble[77]), "doppia", "€ 350 condominio, acqua, riscaldamento e tassa rifiuti inclusi", as.character(mydata$Notas[77]), as.character(mydata$Time[77]), as.character(mydata$Ingresso[77]), as.character(mydata$Soggiorno[77]), as.character(mydata$Cucina[77]), as.character(mydata$Bagno[77]), as.character(mydata$Salone[77]), as.character(mydata$Disimpegno[77]), as.character(mydata$Corridoio[77]), as.character(mydata$Internet[77]), as.character(mydata$Ripostiglio[77]), as.character(mydata$Balcone[77]), as.character(mydata$Termo[77]), as.character(mydata$Terrazzo[77]))  
mydata <- rbind(mydata, new\_row)  
mydata$Habitaciones.Disponibles[77] = "singola"  
mydata$Precio.Mensual[77] = "€ 550 condominio, acqua, riscaldamento e tassa rifiuti inclusi"  
#######################################################################################################################  
new\_row = c(as.character(mydata$Distrito[81]), as.character(mydata$Dirección[81]), as.character(mydata$Tipo.de.Inmueble[81]), "doppia", "€ 325 TUTTO INCLUSO", as.character(mydata$Notas[81]), as.character(mydata$Time[81]), as.character(mydata$Ingresso[81]), as.character(mydata$Soggiorno[81]), as.character(mydata$Cucina[81]), as.character(mydata$Bagno[81]), as.character(mydata$Salone[81]), as.character(mydata$Disimpegno[81]), as.character(mydata$Corridoio[81]), as.character(mydata$Internet[81]), as.character(mydata$Ripostiglio[81]), as.character(mydata$Balcone[81]), as.character(mydata$Termo[81]), as.character(mydata$Terrazzo[81]))  
mydata <- rbind(mydata, new\_row)  
mydata$Habitaciones.Disponibles[81] = "singola"  
mydata$Precio.Mensual[81] = "€ 550 TUTTO INCLUSO"  
#######################################################################################################################  
new\_row = c(as.character(mydata$Distrito[84]), as.character(mydata$Dirección[84]), as.character(mydata$Tipo.de.Inmueble[84]), "doppia", "€ 375 condominio, acqua, riscaldamento tassa rifiuti inclusa", as.character(mydata$Notas[84]), as.character(mydata$Time[84]), as.character(mydata$Ingresso[84]), as.character(mydata$Soggiorno[84]), as.character(mydata$Cucina[84]), as.character(mydata$Bagno[84]), as.character(mydata$Salone[84]), as.character(mydata$Disimpegno[84]), as.character(mydata$Corridoio[84]), as.character(mydata$Internet[84]), as.character(mydata$Ripostiglio[84]), as.character(mydata$Balcone[84]), as.character(mydata$Termo[84]), as.character(mydata$Terrazzo[84]))  
mydata <- rbind(mydata, new\_row)  
mydata$Habitaciones.Disponibles[84] = "singola"  
mydata$Precio.Mensual[84] = "€ 475 condominio, acqua, riscaldamento tassa rifiuti inclusa"  
#######################################################################################################################  
new\_row = c(as.character(mydata$Distrito[85]), as.character(mydata$Dirección[85]), as.character(mydata$Tipo.de.Inmueble[85]), "doppia", "€ 300 spese escluse", as.character(mydata$Notas[85]), as.character(mydata$Time[85]), as.character(mydata$Ingresso[85]), as.character(mydata$Soggiorno[85]), as.character(mydata$Cucina[85]), as.character(mydata$Bagno[85]), as.character(mydata$Salone[85]), as.character(mydata$Disimpegno[85]), as.character(mydata$Corridoio[85]), as.character(mydata$Internet[85]), as.character(mydata$Ripostiglio[85]), as.character(mydata$Balcone[85]), as.character(mydata$Termo[85]), as.character(mydata$Terrazzo[85]))  
mydata <- rbind(mydata, new\_row)  
mydata$Habitaciones.Disponibles[85] = "singola"  
mydata$Precio.Mensual[85] = "€ 450 spese escluse"  
#######################################################################################################################  
mydata$Habitaciones.Disponibles[87] = "singola"  
mydata$Precio.Mensual[87] = "€ 450 condominio, acqua, riscaldamento, tassa rifiuti, internet inclusi"  
#######################################################################################################################  
new\_row = c(as.character(mydata$Distrito[88]), as.character(mydata$Dirección[88]), as.character(mydata$Tipo.de.Inmueble[88]), "doppia", "€ 300 condominio, acqua, riscaldamento, tassa rifiuti inclusi", as.character(mydata$Notas[88]), as.character(mydata$Time[88]), as.character(mydata$Ingresso[88]), as.character(mydata$Soggiorno[88]), as.character(mydata$Cucina[88]), as.character(mydata$Bagno[88]), as.character(mydata$Salone[88]), as.character(mydata$Disimpegno[88]), as.character(mydata$Corridoio[88]), as.character(mydata$Internet[88]), as.character(mydata$Ripostiglio[88]), as.character(mydata$Balcone[88]), as.character(mydata$Termo[88]), as.character(mydata$Terrazzo[88]))  
mydata <- rbind(mydata, new\_row)  
mydata$Habitaciones.Disponibles[88] = "singola"  
mydata$Precio.Mensual[88] = "€ 450 condominio, acqua, riscaldamento, tassa rifiuti inclusi"  
#######################################################################################################################  
mydata$Habitaciones.Disponibles[93] = "singola"  
mydata$Precio.Mensual[93] = "€ 500 TUTTO INCLUSO"  
#######################################################################################################################  
mydata$Habitaciones.Disponibles[99] = "singola"  
mydata$Precio.Mensual[99] = "€ 550 spese escluse"

### Numerizando la Columna de Habitaciones Disponibles

Se le asignó el valor 0 (Simple), 1 (Doble), 2 (Apartamento entero).

#######################################################################################################################  
#Numerizing column Habitaciones Disponibles individual 0, compartida 1, apartamento entero 2,   
mydata$Habitaciones.Disponibles[grepl("(singola)", mydata$Habitaciones.Disponibles)] = 0  
mydata$Habitaciones.Disponibles[grepl("(singole)", mydata$Habitaciones.Disponibles)] = 0  
mydata$Habitaciones.Disponibles[grepl("(Singola)", mydata$Habitaciones.Disponibles)] = 0  
mydata$Habitaciones.Disponibles[grepl("(doppia)", mydata$Habitaciones.Disponibles)] = 1  
mydata$Habitaciones.Disponibles[grepl("(doppie)", mydata$Habitaciones.Disponibles)] = 1  
mydata$Habitaciones.Disponibles[grepl("(letto)", mydata$Habitaciones.Disponibles)] = 1  
mydata$Habitaciones.Disponibles[grepl("(appartamento)", mydata$Habitaciones.Disponibles)] = 2  
mydata$Habitaciones.Disponibles[grepl("(Appartamento)", mydata$Habitaciones.Disponibles)] = 2  
mydata$Habitaciones.Disponibles[grepl("(monolocale)", mydata$Habitaciones.Disponibles)] = 2  
########################################################################################################################

### Añadiendo Otras Características al Dataset, Llenándola y Limpiando la Columna Precio

De la columna **Precio Mensual** se podían extraer características de servicios incluídos en el costo, cómo lo son **Condominio**, **Agua**, **Relleno Sanitario**.

#Adding new features  
mydata$Condominio = 0  
mydata$Acqua = 0  
mydata$tRifiuti = 0  
########################################################################################################################  
#Extracting features for Condominio Column  
mydata$Condominio[grepl("(condominio)", mydata$Precio.Mensual)] = 1  
########################################################################################################################  
#Extracting features for Acqua Column  
mydata$Acqua[grepl("(acqua)", mydata$Precio.Mensual)] = 1  
########################################################################################################################  
#Extracting features for Riscaldamento Column  
mydata$Termo[grepl("(riscaldamento)", mydata$Precio.Mensual)] = 1  
########################################################################################################################  
#Extracting features for Rifiuti Column  
mydata$tRifiuti[grepl("(rifiuti)", mydata$Precio.Mensual)] = 1  
########################################################################################################################  
#Extracting features for TUTTO INCLUSO Column  
mydata$Condominio[grepl("(TUTTO)", mydata$Precio.Mensual)] = 1  
mydata$Acqua[grepl("(TUTTO)", mydata$Precio.Mensual)] = 1  
mydata$tRifiuti[grepl("(TUTTO)", mydata$Precio.Mensual)] = 1  
########################################################################################################################  
#Extracting the price for appartment  
mydata$Precio.Mensual = strsplit(as.character(mydata$Precio.Mensual), ";")  
  
for(i in seq(1:nrow(mydata)))  
{  
 mydata$Precio.Mensual[i] = paste(mydata$Precio.Mensual[[i]], collapse = "")  
   
}  
  
mydata$Precio.Mensual[53] = "€ 425"  
  
mydata$Precio.Mensual = strsplit(as.character(mydata$Precio.Mensual), "\n")  
  
for(i in seq(1:nrow(mydata)))  
{  
 mydata$Precio.Mensual[i] = paste(mydata$Precio.Mensual[[i]], collapse = " ")  
}  
  
mydata$Precio.Mensual = strsplit(as.character(mydata$Precio.Mensual), " ")  
  
for(i in seq(1:nrow(mydata)))  
{  
 mydata$Precio.Mensual[i] = mydata$Precio.Mensual[[i]][2]  
}  
#########################################################################################################################  
#Deleting Disctrict Column   
mydata$Distrito = NULL

### Añadiendo la Función de Pesos Ajustadas a las Necesidades

Para poder decir cuál es la mejor vivienda para ela migo de mi amigo se le dió peso a cada una de las características del hogar, se sumaron y se dividieron entre el precio total de la vivienda, dando así la vivienda más prometedora a elegir.

############################################################################################################################  
#Adding function column values  
mydata$Function = 0  
mydata$Function = as.numeric(mydata$Function)  
############################################################################################################################  
#Assigning weigths to variable Tipo de inmueble  
#If it is a Apparment is 100  
# If it is an mini apparment if 75  
# If it is an flat is 50  
mydata$Tipo.de.Inmueble = as.character(mydata$Tipo.de.Inmueble)  
mydata$Tipo.de.Inmueble[mydata$Tipo.de.Inmueble == "0" ] = "100"  
mydata$Tipo.de.Inmueble[mydata$Tipo.de.Inmueble == "1" ] = "75"  
mydata$Tipo.de.Inmueble[mydata$Tipo.de.Inmueble == "2" ] = "50"  
mydata$Tipo.de.Inmueble = as.numeric(mydata$Tipo.de.Inmueble)  
############################################################################################################################  
#Assigning weigths to variable Habitaciones Disponible  
#If it is singola is 75  
# If it a doppie is 25  
# If it is complete is 100  
mydata$Habitaciones.Disponibles = as.character(mydata$Habitaciones.Disponibles)  
mydata$Habitaciones.Disponibles[mydata$Habitaciones.Disponibles == "0" ] = "75"  
mydata$Habitaciones.Disponibles[mydata$Habitaciones.Disponibles == "1" ] = "25"  
mydata$Habitaciones.Disponibles[mydata$Habitaciones.Disponibles == "2" ] = "100"  
mydata$Habitaciones.Disponibles = as.numeric(mydata$Habitaciones.Disponibles)  
############################################################################################################################  
#Assigning weigths to variable Ingresso  
#If it has is 100  
#Else is 50  
mydata$Ingresso= as.character(mydata$Ingresso)  
mydata$Ingresso[grepl("(0)", mydata$Ingresso)] = "50"  
mydata$Ingresso[grepl("(1)", mydata$Ingresso)] = "100"  
mydata$Ingresso= as.numeric(mydata$Ingresso)  
#############################################################################################################################  
#Assigning weigths to variable soggiorno  
#If it has is 100  
#Else is 50  
mydata$Soggiorno = as.character(mydata$Soggiorno)  
mydata$Soggiorno[grepl("(0)", mydata$Soggiorno)] = "50"  
mydata$Soggiorno[grepl("(1)", mydata$Soggiorno)] = "100"  
mydata$Soggiorno = as.numeric(mydata$Soggiorno)  
############################################################################################################################  
#Assigning weigths to variable cucina  
#If it has is 100  
#Else is 25  
mydata$Cucina = as.character(mydata$Cucina)  
mydata$Cucina[grepl("(0)", mydata$Cucina)] = "25"  
mydata$Cucina[grepl("(1)", mydata$Cucina)] = "100"  
mydata$Cucina = as.numeric(mydata$Cucina)  
############################################################################################################################  
#Assigning weigths to variable bagno  
#If it has is 100  
#Else is 0  
mydata$Bagno = as.numeric(mydata$Bagno)  
mydata$Bagno = mydata$Bagno \* 100  
############################################################################################################################  
#Assigning weigths to variable salone  
#If it has is 50  
#Else is 25  
mydata$Salone = as.character(mydata$Salone)  
mydata$Salone[grepl("(0)", mydata$Salone)] = "25"  
mydata$Salone[grepl("(1)", mydata$Salone)] = "50"  
mydata$Salone = as.numeric(mydata$Salone)  
############################################################################################################################  
#Assigning weigths to variable disimpegno  
#If it has is 50  
#Else is 25  
mydata$Disimpegno = as.character(mydata$Disimpegno)  
mydata$Disimpegno[grepl("(0)", mydata$Disimpegno)] = "25"  
mydata$Disimpegno[grepl("(1)", mydata$Disimpegno)] = "50"  
mydata$Disimpegno = as.numeric(mydata$Disimpegno)  
############################################################################################################################  
#Assigning weigths to variable corridoio  
#If it has is 50  
#Else is 25  
mydata$Corridoio = as.character(mydata$Corridoio)  
mydata$Corridoio[grepl("(0)", mydata$Corridoio)] = "25"  
mydata$Corridoio[grepl("(1)", mydata$Corridoio)] = "50"  
mydata$Corridoio = as.numeric(mydata$Corridoio)  
############################################################################################################################  
#Assigning weigths to variable internet  
#If it has is 75  
#Else is 25  
mydata$Internet = as.character(mydata$Internet)  
mydata$Internet[grepl("(0)", mydata$Internet)] = "0"  
mydata$Internet[grepl("(1)", mydata$Internet)] = "100"  
mydata$Internet = as.numeric(mydata$Internet)  
############################################################################################################################  
#Assigning weigths to variable ripostiglio  
#If it has is 50  
#Else is 25  
mydata$Ripostiglio = as.character(mydata$Ripostiglio)  
mydata$Ripostiglio[grepl("(0)", mydata$Ripostiglio)] = "25"  
mydata$Ripostiglio[grepl("(1)", mydata$Ripostiglio)] = "50"  
mydata$Ripostiglio = as.numeric(mydata$Ripostiglio)  
############################################################################################################################  
#Assigning weigths to variable balcone  
#If it has is 50  
#Else is 25  
mydata$Balcone = as.character(mydata$Balcone)  
mydata$Balcone[grepl("(0)", mydata$Balcone)] = "25"  
mydata$Balcone[grepl("(1)", mydata$Balcone)] = "50"  
mydata$Balcone = as.numeric(mydata$Balcone)  
############################################################################################################################  
#Assigning weigths to variable termo  
#If it has is 100  
#Else is 0  
mydata$Termo = as.character(mydata$Termo)  
mydata$Termo[grepl("(0)", mydata$Termo)] = "0"  
mydata$Termo[grepl("(1)", mydata$Termo)] = "100"  
mydata$Termo = as.numeric(mydata$Termo)  
############################################################################################################################  
#Assigning weigths to variable terrazo  
#If it has is 50  
#Else is 25  
mydata$Terrazzo = as.character(mydata$Terrazzo)  
mydata$Terrazzo[grepl("(0)", mydata$Terrazzo)] = "25"  
mydata$Terrazzo[grepl("(1)", mydata$Terrazzo)] = "50"  
mydata$Terrazzo = as.numeric(mydata$Terrazzo)  
############################################################################################################################  
#Assigning weigths to variable condominio  
#If it has is 100  
#Else is 0  
mydata$Condominio = as.character(mydata$Condominio)  
mydata$Condominio[grepl("(0)", mydata$Condominio)] = "0"  
mydata$Condominio[grepl("(1)", mydata$Condominio)] = "100"  
mydata$Condominio = as.numeric(mydata$Condominio)  
############################################################################################################################  
#Assigning weigths to variable acqua  
#If it has is 100  
#Else is 0  
mydata$Acqua = as.character(mydata$Acqua)  
mydata$Acqua[grepl("(0)", mydata$Acqua)] = "0"  
mydata$Acqua[grepl("(1)", mydata$Acqua)] = "100"  
mydata$Acqua = as.numeric(mydata$Acqua)  
############################################################################################################################  
#Assigning weigths to variable tassa rifiuti  
#If it has is 100  
#Else is 0  
mydata$tRifiuti = as.character(mydata$tRifiuti)  
mydata$tRifiuti[grepl("(0)", mydata$tRifiuti)] = "0"  
mydata$tRifiuti[grepl("(1)", mydata$tRifiuti)] = "50"  
mydata$tRifiuti = as.numeric(mydata$tRifiuti)  
############################################################################################################################  
#Filling function column  
mydata$Function = as.numeric(mydata$Function)  
mydata$Function = mydata$Function - as.numeric(mydata$Time)  
  
for(i in seq(1:(ncol(mydata)-1)))  
{  
 if( (i != 1) && (i != 5) && (i != 4) && (i != 6) )  
 {  
 mydata$Function =mydata$Function + mydata[i]  
 }  
}  
mydata$Function = mydata$Function / as.numeric(mydata$Precio.Mensual)

## Dividiendo los Conjuntos de Datos en Niños y Niñas Para Hacer un Modelo Independiente

Se pidió que se hiciera un modelo independiente para niños y niñas, por lo tanto se separó el dataset original en dos.

############################################################################################################################  
#Dividing datasets into boys and girls  
boys = subset.data.frame(mydata, Notas == "0" | Notas == "2")  
girls = subset.data.frame(mydata, Notas == "1" | Notas == "2")  
############################################################################################################################

## Eligiendo la Mejor Vivienda Para Niños

Se eligió aquella dirección que tuviese un mayor valor en la función creada anteriormente

#Choosing the best place for boys  
max\_boys = max(boys$Function)  
boys$Dirección[which.max(as.numeric(unlist(boys$Function)))]

## [1] "via Giovanni Conti"

## Eligiendo la Mejor Vivienda Para Niñas

Se eligió aquella dirección que tiviese un mayor valor en la función creada anteriormente

#Choosing the best place for girls  
max\_girls = max(girls$Function)  
girls$Dirección[which.max(as.numeric(unlist(girls$Function)))]

## [1] "Via dei Georgofili"

## Regresión Lineal

Se usó regresión lineal para estimar el precio de una vivienda basándose en sus características.

### Modelo Para Niños

#### Creando el Dataframe

Se creó el dataframe y se eiliminaron las columnas que no usaremos para predecir o que no son numéricas y no s epueden normalizar

#Linear Regression Section for Boys  
regression\_boys = boys  
regression\_boys$Precio.Mensual = NULL  
regression\_boys$Dirección = NULL  
regression\_boys$Function = NULL  
regression\_boys$Time = NULL  
############################################################################################################################  
#Normalizing columns  
for(i in seq(1:ncol(regression\_boys)))  
{  
 regression\_boys[i] = as.numeric(unlist(regression\_boys[i]))  
}  
regression\_boys = as.data.frame(lapply(regression\_boys[1:ncol(regression\_boys)], normalize))  
#Adding column to be predicetd  
regression\_boys$Precio.Mensual = boys$Precio.Mensual  
regression\_boys$Precio.Mensual = as.numeric(regression\_boys$Precio.Mensual)

#### Dividiendo los Conjuntos en Training y Testing

Se realizó un sampling colocando el 80% de la muestra en el training y el restante en el testing.

#Splitting dataset into training and testing  
set.seed(777)  
sub = sample(x = nrow(regression\_boys), size = floor(nrow(regression\_boys) \* 0.8), replace = F)  
training\_regression\_boys = regression\_boys[sub, ]  
testing\_regression\_boys = regression\_boys[-sub, ]

#### Aplicando la Regresión Lineal

Predeciremos el precio basándonos en las demás características del inmueble.

# Applying Regression  
model\_boys = lm(training\_regression\_boys$Precio.Mensual ~ ., training\_regression\_boys)

#### Probando el Modelo

Se prueba el modelo con el training.

# Testing the model  
testing\_regression\_boys$Estimado = predict(model\_boys, newdata = testing\_regression\_boys)

#### Mostrando los resultados

Los resultados se pueden interpretar de la siguiente manera:

1. Si las casas tienen precios similares, entonces el precio está proporcional a suu valor.
2. Si el valor real es mayor es mayor a la prediccón, entonces esa casa está sobrevalorada.
3. Si el valor real es menor a la predicción, entonces es una beuna opción ofertar por esa casa.

# Showing reslst  
result\_boys = as.data.frame(testing\_regression\_boys$Estimado)  
result\_boys$Real = testing\_regression\_boys$Precio.Mensual  
result\_boys

## testing\_regression\_boys$Estimado Real  
## 1 648.8714 550  
## 2 958.8666 700  
## 3 517.8889 475  
## 4 463.3078 380  
## 5 698.7958 300  
## 6 704.5439 350  
## 7 559.2103 1200  
## 8 734.7247 850  
## 9 818.0739 800  
## 10 588.2402 850  
## 11 259.2261 340  
## 12 510.0080 475  
## 13 529.0676 450  
## 14 614.6511 900  
## 15 514.9346 650

### Modelo para niñas

#### Creando el Dataframe

Se creó el dataframe y se eiliminaron las columnas que no usaremos para predecir o que no son numéricas y no s epueden normalizar.

#Linear Regression Section for girls  
regression\_girls = girls  
regression\_girls$Precio.Mensual = NULL  
regression\_girls$Dirección = NULL  
regression\_girls$Function = NULL  
regression\_girls$Time = NULL  
############################################################################################################################  
#Normalizing columns  
for(i in seq(1:ncol(regression\_girls)))  
{  
 regression\_girls[i] = as.numeric(unlist(regression\_girls[i]))  
}  
regression\_girls = as.data.frame(lapply(regression\_girls[1:ncol(regression\_girls)], normalize))  
#Adding column to be predicetd  
regression\_girls$Precio.Mensual = girls$Precio.Mensual  
regression\_girls$Precio.Mensual = as.numeric(regression\_girls$Precio.Mensual)

#### Dividiendo los Conjuntos en Training y Testing

Se realizó un sampling colocando el 80% de la muestra en el training y el restante en el testing.

#Splitting dataset into training and testing  
set.seed(777)  
sub = sample(x = nrow(regression\_girls), size = floor(nrow(regression\_girls) \* 0.8), replace = F)  
training\_regression\_girls = regression\_girls[sub, ]  
testing\_regression\_girls = regression\_girls[-sub, ]

#### Aplicando la Regresión Lineal

Predeciremos el precio basándonos en las demás características del inmueble.

# Applying the model  
model\_girls = lm(training\_regression\_girls$Precio.Mensual ~ ., training\_regression\_girls)

#### Probando el Modelo

Se prueba el modelo con el training.

# Testing the model  
testing\_regression\_girls$Estimado = predict(model\_girls, newdata = testing\_regression\_girls)

#### Mostrando los resultados

Los resultados se pueden interpretar de la siguiente manera:

1. Si las casas tienen precios similares, entonces el precio está proporcional a suu valor.
2. Si el valor real es mayor es mayor a la prediccón, entonces esa casa está sobrevalorada.
3. Si el valor real es menor a la predicción, entonces es una beuna opción ofertar por esa casa.

# Showing results  
result\_girls = as.data.frame(testing\_regression\_girls$Estimado)  
result\_girls$Real = testing\_regression\_girls$Precio.Mensual  
result\_girls

## testing\_regression\_girls$Estimado Real  
## 1 755.1646 650  
## 2 546.3297 250  
## 3 676.3008 1300  
## 4 538.8436 450  
## 5 604.1777 430  
## 6 609.9173 475  
## 7 414.3919 300  
## 8 467.4659 350  
## 9 575.4404 350  
## 10 474.6497 420  
## 11 535.7482 450  
## 12 500.3736 340  
## 13 529.2138 410  
## 14 467.4659 450  
## 15 511.8528 450  
## 16 191.9640 340  
## 17 253.4823 325  
## 18 480.3894 450  
## 19 540.5901 530  
## 20 519.5743 650  
## 21 546.3297 550