ITESO

Boosting

PAP Modelación Matemática para el desarrollo de planes y proyectos de negocio

Sara Eugenia Rodríguez Reyes

19/09/2015

Introducción

Los sistemas de intercambio de bicicletas son la nueva generación de alquiler de bicicletas tradicionales, donde todo el proceso de registro, renta y devolución se ha convertido en automático. A través de estos sistemas, el usuario puede alquilar una bicicleta en una posición particular y devolverla en otra posición. Actualmente, hay alrededor de más de 500 programas de intercambio de bicicletas en todo el mundo con más de 500 mil bicicletas. Hoy en día, existe un gran interés en estos sistemas debido a su importante papel en cuestiones de tráfico, ambientales y de salud.

Además de interesantes aplicaciones del mundo real de los sistemas de intercambio de bicicletas, las características de los datos que se generan por estos sistemas los hacen atractivos para la investigación. A diferencia de otros servicios de transporte como el autobús o el metro, la duración de los viajes, la posición de salida y de llegada se registran en estos sistemas. Por lo tanto, se espera que la mayoría de los eventos importantes de la ciudad se puedan detectar a través del seguimiento de estos datos.

La base de datos utilizada se llama “Bike Sharing Dataset”, obtenida de UCI Machine Learning Repository. El conjunto de datos contiene la hora de alquiler de bicicletas entre los años 2011 y 2012 con su correspondiente clima y la información de temporada.

Se tienen 15,000 filas y 17 columnas, las últimas descritas a continuación:

1. Instant: índice de registro
2. Dteday: fecha
3. Season: temporada (1: primavera, 2: verano, 3: otoño, 4: invierno)
4. Yr: año
5. Mnth: mes
6. Hr: hora
7. Holiday: si es día de asueto o no
8. Weekday: día de la semana
9. Workingday: si no es fin de semana ni vacaciones es 1, de otra manera es 0
10. Weathersit:
    a. 1. Claro, pocas nubes, medio nublado
    b. 2. Bruma, nublado, niebla, cielo nuboso
    c. 3. Poca nieve, lluvia ligera, nubes dispersas
    d. 4. Lluvia pesada, nieve, niebla
11. Temp: la temperatura normalizada en grados Celsius
12. Atemp: temperatura normalizada de sensación en grados Celsius
13. Hum: humedad normalizada

14. Windspeed: velocidad del viento normalizada
15. Casual: cuenta de los usuarios ocasionales
16. Registered: cuenta de los usuarios registrados
17. Cnt: recuento total de las bicicletas alquiladas incluyendo usuarios casuales y registrados

La variable a predecir es "cnt", es decir el número de bicicletas rentadas por hora, utilizando todas las variables con excepción del número de registro, la fecha, las bicicletas rentadas con una cuenta y las bicicletas rentadas ocasionalmente.

Boosted Trees

Bagging es crear muchas copias del conjunto de entrenamiento original, ajustando un árbol de decisión por separado a cada copia y luego combinando todos los árboles para crear un solo modelo de predicción. Cada árbol se construye con cada conjunto de datos, independiente de otros árboles.

Boosting es parecido, solo que cada árbol se construye usando información de árboles anteriores. Cada árbol es ajustado en una versión modificada del conjunto de datos original.

En este método, ajustamos un árbol usando los residuales actuales. A continuación se añade este nuevo árbol de decisión a una función ajustada con el fin de actualizar los residuales. Cada uno de estos árboles puede ser bastante pequeño, con pocos nodos terminales, determinados por el parámetro d en el algoritmo. Al ajustar árboles pequeños a los residuales, se va mejorando las áreas donde no funciona bien.

El parámetro de contracción λ hace más lento el proceso, permitiendo que diferentes formas de árboles ataquen los residuales.

Boosting necesita tres parámetros:

1. El número de árboles B. Se puede sobre ajustar el modelo si es muy grande.
2. El parámetro de contracción λ, un número pequeño y positivo. Éste controla la tasa con la que el método aprende. Si es un número demasiado pequeño puede requerir usar un valor grande de B para obtener un buen resultado.
3. El número de particiones d en cada árbol, el cual controla la complejidad; es la profundidad de la interacción.

Boosting

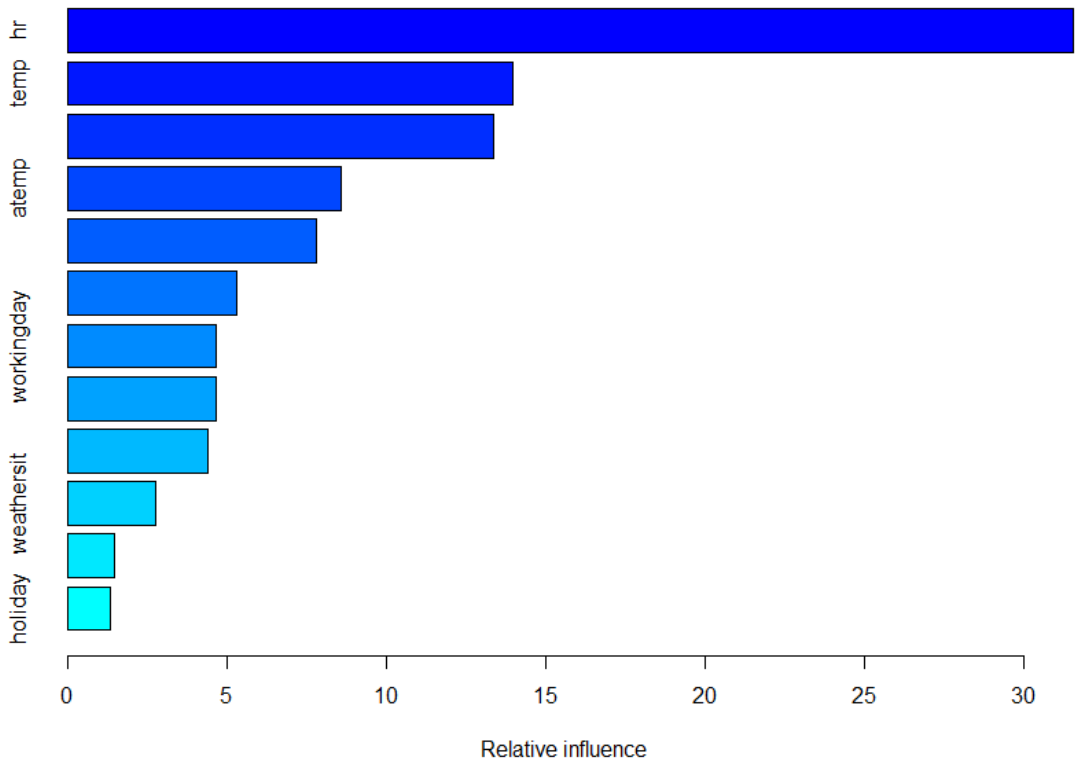
```
##Establecer directorio de trabajo
setwd("C:/Users/Sara/Documents/ITESO/PAP2/Boosting")
##Cargar datos
Bikes <- read.table("./hour.csv", sep=",", header=TRUE)
train2 <- read.table("./train.csv", sep=",", header=TRUE)
##Cargar librerías
library(ggplot2)   #Para cargar gráficos
library(lubridate) #Para trabajar con fechas y tiempos
library(dplyr)     #Para trabajar con objetos data.frame
library(scales)    #Escalas de las gráficas
library(gbm)       #Bossted Trees
library(plyr)
library(randomForest)    #Random Forest
library(readr)

##Tomar muestra de entrenamiento y prueba
train <- sample (1: nrow(Bikes ), nrow(Bikes ) / 2)
test <- Bikes [-train ,"cnt"]

##Utilizar método de boosting trees, hacer predicciones, probar con diferentes
parámetros
##Minimizar el error con los parámetros óptimos
set.seed(1)
ntree <- c(500,1000,2000,5000)
interact <- seq(1,4,1)
lambda <- seq(0.25,1,0.25)
error <- array(dim = c(4,4,4))
for (i in 1:length(ntree)){
  for(j in 1:length(interact)){
    for (k in 1:length(lambda)){
      BoostBikes <- gbm(cnt ~. -instant -dteday -casual -registered, data =
Bikes [train ,], distribution = "gaussian",                          n.trees =
ntree[i], interaction.depth = interact[j], shrinkage = lambda[k])
      yhatBoost    <-    predict(BoostBikes,    newdata=Bikes    [-train    ,],
n.trees=ntree[i])
      mse <- mean((yhatBoost-test)^2)
      error[i,j,k] <- mse
    }
  }
}
```

El error mínimo es utilizando los parámetros:

- Ntree: 2000
- Interact.depth: 4
- Lambda: 0.25

```
summary (BoostBikes)
```

```
                  var   rel.inf
hr                 hr 31.564028
temp             temp 13.972176
hum               hum 13.360277
atemp           atemp  8.608814
windspeed   windspeed  7.802752
mnth             mnth  5.335257
workingday workingday  4.676460
weekday       weekday  4.651189
yr                 yr  4.425730
weathersit weathersit  2.755786
season         season  1.478523
holiday       holiday  1.369009
```



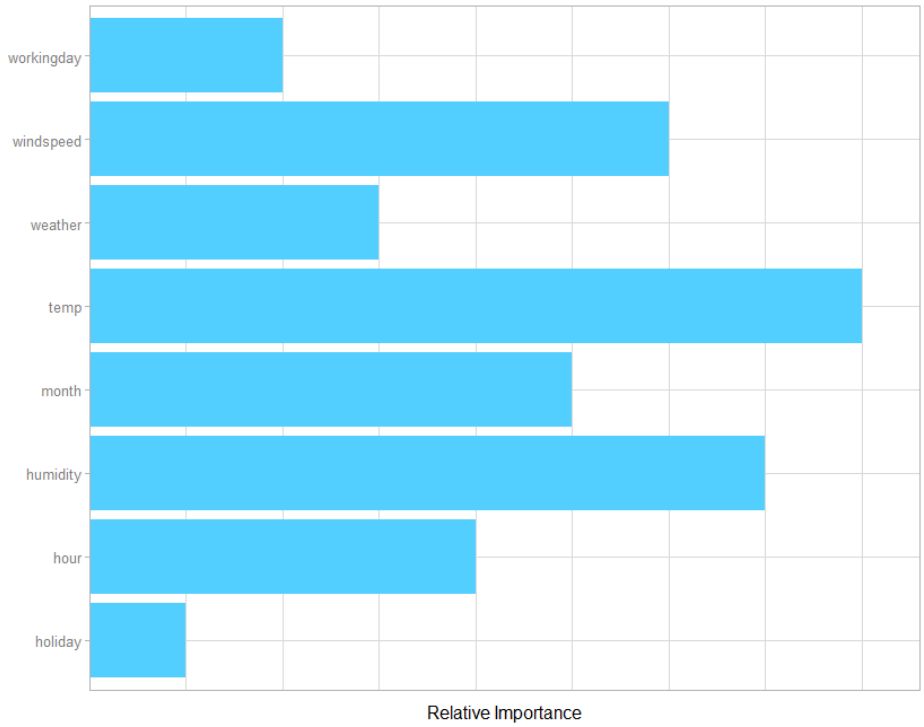
Las variables más significativas son “hr”, “hum” y “temp” que son la hora en que son rentadas las bicicletas, la humedad y la temperatura de esa hora respectivamente.

Partial Dependence Plot Boosting

Las gráficas de dependencia parciales muestran la dependencia entre la función objetivo y un conjunto de características. Debido a los límites de la percepción humana, el tamaño del conjunto de características objetivo debe ser pequeño por lo tanto estas características suelen elegirse entre las más importantes.

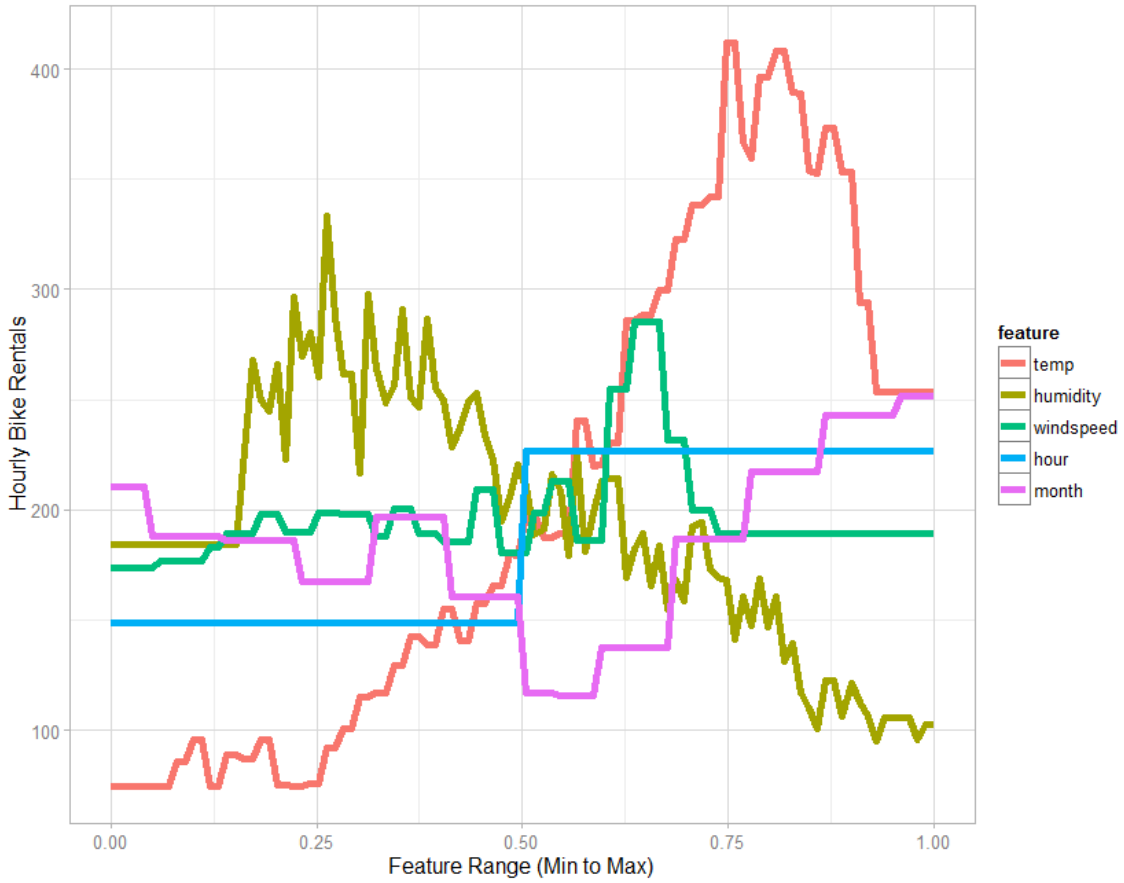
A continuación se muestran las gráficas de dependencia parcial de las variables más importantes del modelo para Boosted Trees.

```
sample_locs <- sample(nrow(train2)) #muestra de prueba
entrenamiento <- as.integer(nrow(train2)*0.7)  #muestra de entrenamiento
entrInterno<-train2[sample_locs[1:entrenamiento],]#submuestra ntrenamiento
entrValidacion   <-   train2[sample_locs[(entrenamiento+1):nrow(train2)],]
#muestra de prueba

#funcion para extraer caracteristicas de las variables
extract_features <- function(data) {
  caracteristicas   <-   c("holiday",   "workingday",   "weather",   "temp",
"humidity", "windspeed","hour","month","count")
  data$hour  <- hour(ymd_hms(data$datetime))
  data$month <- month(ymd_hms(data$datetime))
  return(data[,caracteristicas])
}
caracteristicas <- extract_features(entrInterno)

#Boosting utilizando parámetros optimos calculados anteriormente
BoostBikes1  <-  gbm(count  ~.  ,  data  =  caracteristicas,  distribution  =
"gaussian", n.trees = 2000, interaction.depth = 4, shrinkage = 0.25)
#Importancia de las variables
imp1 <- as.matrix(summary(BoostBikes1))
featureImportance1          <-          data.frame(Feature=row.names(imp1),
Importance=imp1[,2])
#Grafica de la importancia por variable
ggplot(featureImportance1,       aes(x=reorder(Feature,       Importance),
y=Importance)) +
  geom_bar(stat="identity", fill="#53cfff") +
  coord_flip() +
  theme_light(base_size=16) +
  xlab("") +
  ylab("Relative Importance") +
  theme(plot.title   = element_text(size=18),
        strip.text.x = element_blank(),
        axis.text.x  = element_blank(),
        axis.ticks.x = element_blank())
```



Con esos parámetros de Boosted Trees, las variables más importantes son: temp, humidity y windspeed.

```
#Partial Dependence Plot
partials1 <- data.frame()

for (i in seq_along(names(caracteristicas[,-9]))) {
  partial1<-plot(BoostBikes1,names(caracteristicas)[i], return.grid =TRUE)
  xt <- rescale(partial1[,1])
  partials1<-rbind(partials1,data.frame(x=partial1[,1],xt=xt,
y=partial1$y, feature=names(caracteristicas)[i]))
}

#Ajustar rangos da las caracteristicas
ranges <- ddply(partials1, "feature", function(d) {
  r <- range(d$y)
  data.frame(feature=d$feature[1], range=r[2]-r[1])
})

features_to_plot <- ranges[ranges$range>0.05*max(ranges$range),"feature"]

#Graficas
ggplot(partials1[partials1$feature %in% features_to_plot,], aes(x=xt, y=y,
color=feature)) +
  geom_line(size=2) +
  theme_light(base_size=16) +
  xlab("Feature Range (Min to Max)") +
  ylab("Hourly Bike Rentals")
```

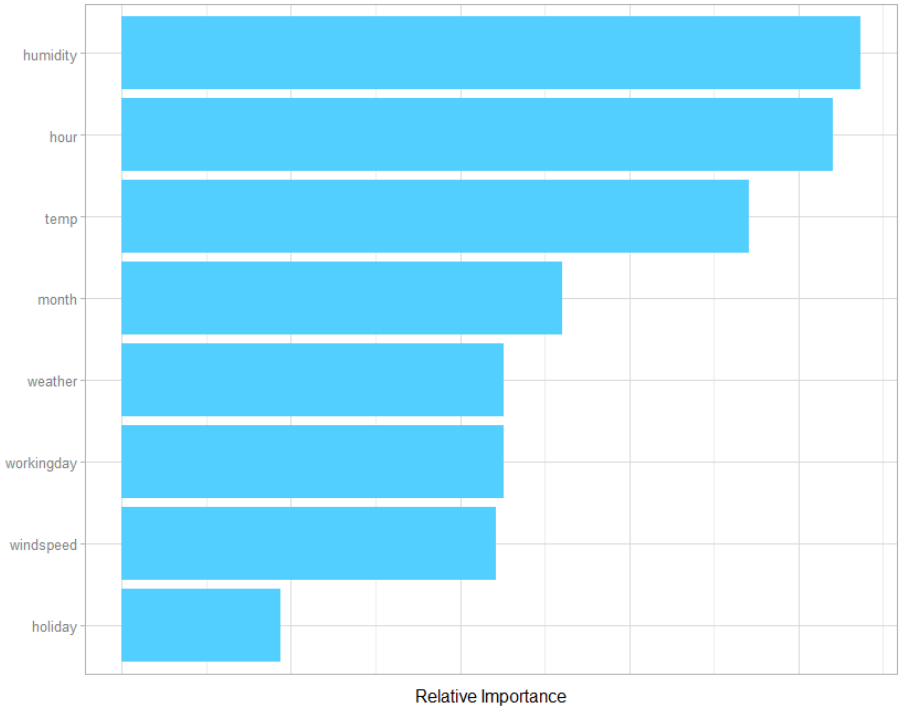


Se puede apreciar que se rentan más bicicletas cuando la temperatura es más alta durante el día. Contrariamente, la renta de bicicletas disminuye cuando se incrementa la humedad.

Comparación con Random Forest

Se tiene información sobre las fechas, clima, horarios de la renta por hora de bicicletas. Se quiere determinar qué factores predicen la demanda de renta de bicicletas; de la misma forma se busca entender cómo estos factores afectan la demanda. Para hacer esto, se entrenará un modelo de Random Forest.

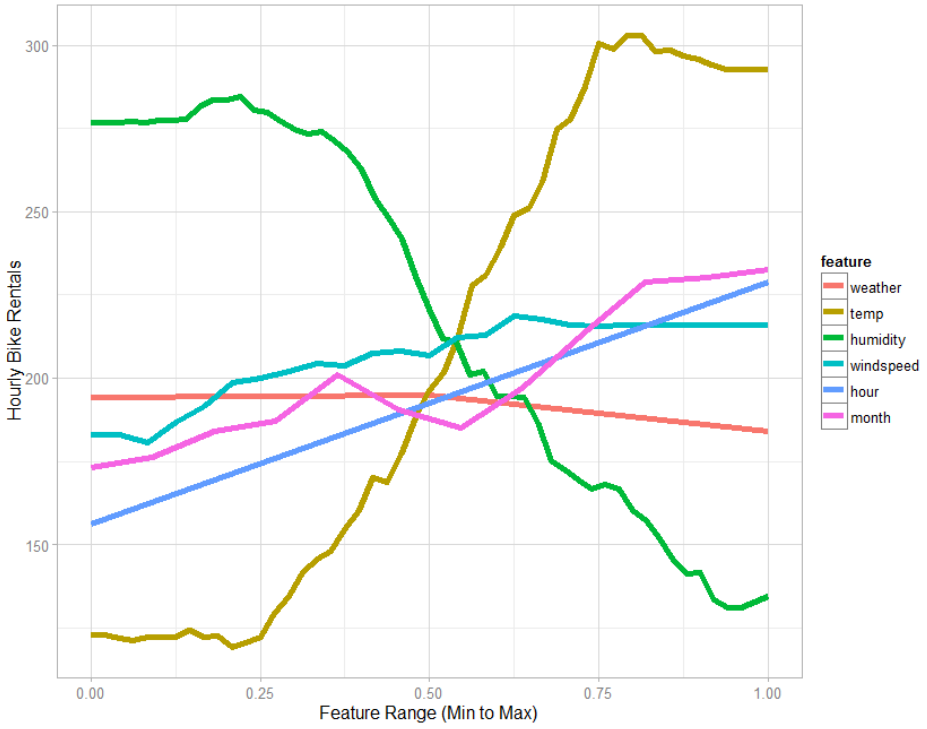
```
#Random Forest
rf<-
randomForest(caracteristicas,entrInterno$count,ntree=100,importance=TRUE)
#De este modelo, calculamos la importancia de las características
imp <- importance(rf, type=1)
featureImportance           <-           data.frame(Feature=row.names(imp),
Importance=imp[,1])
#Gráfica de importancia
ggplot(featureImportance,        aes(x=reorder(Feature,        Importance),
y=Importance)) +
  geom_bar(stat="identity", fill="#53cfff") +
  coord_flip() +
  theme_light(base_size=16) +
  xlab("") +
  ylab("Relative Importance") +
  theme(plot.title    = element_text(size=18),
        strip.text.x = element_blank(),
        axis.text.x  = element_blank(),
        axis.ticks.x = element_blank())
```



Partial Dependence Random Forest

Podemos usar el modelo random forest para general partial plots de cada característica, permitiendo visualizar lo que el modelo ha aprendido de ellas

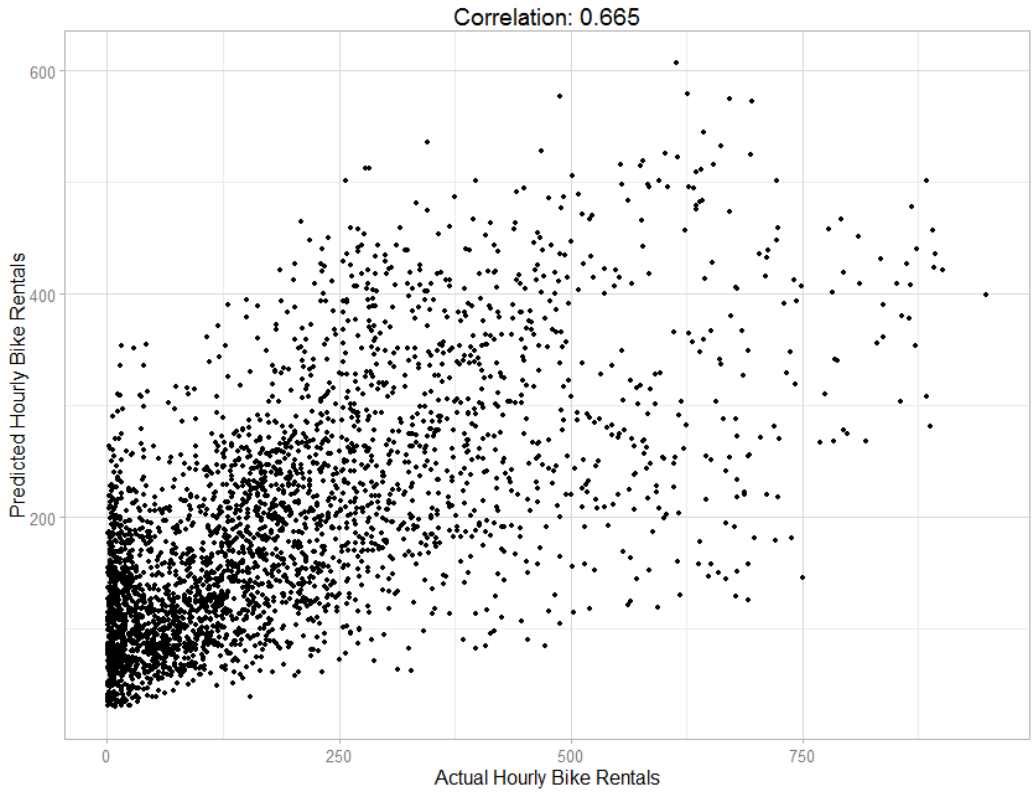
```
partials <- data.frame()

for (i in seq_along(names(caracteristicas))) {
  partial <- partialPlot(rf, caracteristicas, names(caracteristicas)[i],
plot=FALSE)
  xt <- rescale(partial$x)
  partials <- rbind(partials, data.frame(x=partial$x, xt=xt, y=partial$y,
feature=names(caracteristicas)[i]))
}

ranges <- ddply(partials, "feature", function(d) {
  r <- range(d$y)
  data.frame(feature=d$feature[1], range=r[2]-r[1])
})

features_to_plot <- ranges[ranges$range>0.05*max(ranges$range),"feature"]

ggplot(partials[partials$feature %in% features_to_plot,], aes(x=xt, y=y,
color=feature)) +
  geom_line(size=2) +
  theme_light(base_size=16) +
  xlab("Feature Range (Min to Max)") +
  ylab("Hourly Bike Rentals")
```



Al igual que Boosted trees, la renta de bicicletas aumenta cuando la temperatura se incrementa. La renta disminuye mientras que la humedad se reduce.

Al hacer las predicciones en nuestro set de validación, vemos que las correlaciones son altas.

```
valid_features <- extract_features(entrValidacion)
valid_features$Predictions                  <-                  predict(rf,
extract_features(entrValidacion))
valid_features$Actuals     <- entrValidacion$count

ggplot(valid_features, aes(x=Actuals, y=Predictions)) +
  geom_point() +
  theme_light(base_size=16) +
  xlab("Actual Hourly Bike Rentals") +
  ylab("Predicted Hourly Bike Rentals") +
  ggtitle(paste0("Correlation:     ",     round(cor(valid_features$Actuals,
valid_features$Predictions), 3)))
```



Conclusiones

Utilizando *Boosted Trees*, para predecir la cantidad de bicicletas que se van a rentar; hay que tomar en cuenta principalmente tres aspectos:

1. Temperatura
2. Humedad
3. Hora

Por lo tanto se recomienda que la empresa que implemente un sistema de renta de bicicletas sistematizado tome en cuenta el clima de la región donde se va a implementar dicho programa así como la jornada laboral de los trabajadores en dicha región.