Taller de ML: práctica

Pablo Hidalgo

Entrega: 21/10/2019

Esta práctica consiste en la aplicación de los contenidos que se han visto en el taller de ML. **La fecha límite de entrega es el 21 de octubre.** La entrega deberá consistir en

- **script de R** donde aparezca todo el código empleado para realizar la práctica. Es recomendable utilizar comentarios para aclarar el código (recuerda, para escribir un comentario empieza por la línea `#`),
- **documento en pdf** donde aparezcan los gráficos, resultados y se hagan los comentarios apropiados. Se puede hacer en cualquier editor de texto (por ejemplo, word) y luego exportase a pdf.

Los datos

Para esta entrega vamos a utilizar unos datos muy similares a los vistos en clase. Son datos de **ventas de casas** pero de otra zona de Estados Unidos, en concreto del Condado de King (Washington).

Las variables de este conjunto de datos son:

- `id`: identificador único de cada casa vendida.
- `date`: fecha de venta.
- `price`: precio de venta (en dólares).
- `bedrooms`: número de habitaciones.
- `bathrooms`: número de baños donde 0.5 significa una habitación con aseo (baño sin ducha).
- `sqft_living`: área habitable de la casa (en pies cuadrados).
- `sqft_lot`: área de la parcela (en pies cuadrados).
- `floors`: número de plantas de la vivienda.
- `waterfront`: variable binaria (*dummy*) indicando si la vivienda tiene vistas al mar o no.
- `view`: índice (0-4) de las vistas.
- `condition`: índice (1-5) del estado de la vivienda.
- `grade`: índice (1-13) de la calidad de la construcción y diseño.
- `sqft_above`: área de la parte que está por encima del nivel del suelo (en pies cuadrados).
- `sqft_basement`: área del sótano (en pies cuadrados).
- `yr_built`: año de construcción.
- `yr_renovated`: año de la última reforma.
- `zipcode`: código postal.
- `lat`: latitud.
- `long`: longitud.
- `sqft_living15`: área de vivienda media de las 15 casas más próximas (en pies cuadrados).
- `sqft_lot15`: área de parcela media de las 15 casas más próximas (en pies cuadrados).

Desarrollo de la práctica.

Comienza cargando el archivo `.csv` en la memoria de R. Para ello, recuerda que puedes utilizar la función `read_csv()` (previamente, deberías haber cargado la librería `tidyverse`).

Inspecciona las variables del conjunto de datos mediante la función `skim()` del paquete `skimr`. Comenta el resultado destacando aquello que encuentres más relevante.

Al tratarse, de nuevo, de un conjunto de datos de Estados Unidos, las variables de superficie vienen expresadas en pies cuadrados. **Convierte todas las variables de superficie a metros cuadrados.**

Estos datos contienen la ubicación (latitud y longitud) de cada venta. Existen paquetes específicos para poder visualizar esta información. **Instala el paquete `leaflet` y ejecuta la siguiente sentencia:**

```
library(leaflet)
# Susituye nombre_datos por el nombre que tengan tus datos
leaflet(nombre_datos) %>% addTiles() %>% addCircleMarkers()
```

Representa la relación que hay entre la superificie de cada vivienda y su precio con un gráfico de puntos. Cambia el título y el nombre de los ejes para hacerlo más fácil de leer.

Divide el conjunto de datos en `train` y `test` de forma que sean el 80% y el 20% de los datos, respectivamente.

Justifica qué variables del conjunto de datos debería excluirse de un modelo para predecir el precio de venta.

Ajusta primero un modelo de regresión lineal utilizando solamente la superficie de la casa para predecir el precio de venta (llama a este modelo `lm1`). Entrena otro modelo utilizando la superficie y la variable `grade` (llámalo `lm2`). **Comenta ambos modelos y sus diferencias.**

Ejecuta las siguientes líneas de código:

```
#Cambia datos_test por el nombre que le hayas dado a test
pred_test <- select(datos_test, price)
pred_test$pred_lm1 <- predict(lm1, newdata = datos_test)
pred_test$pred_lm2 <- predict(lm2, newdata = datos_test)
rmse <- function(price, pred) sqrt(mean((price - pred)^2))
error <- tibble(
  modelo = c("lm1", "lm2"),
  error = c(rmse(pred_test$price, pred_test$pred_lm1), rmse(pred_test$price, pred_test$pred_lm2))
)
```

Representa el gráfico de residuos de ambos modelos y coméntalos.

En clase hemos visto el modelo de Gradient Boosting, uno de los modelos de *machine learning* más utilizados. Ejecuta el siguiente código cambiando lo necesario:

```
train_x <- select(datos_train,
                  -nombre_variable_precio,
                  -nombre_variable_id,
                  -nombre_variable_fecha) %>%
  as.matrix()

test_x <- select(datos_test,
                 -nombre_variable_precio,
                 -nombre_variable_id,
                 -nombre_variable_fecha) %>%
  as.matrix()
test_x[is.na(test_x)] <- 0


library(xgboost)
```

```
gradient_boost <- xgboost(data = train_x,
                          label = datos_train$nombre_variable_precio,
                          nrounds = 1000,
                          params = list(eta = 0.01,
                                        max_depth = 3,
                                        colsample_bytree = 0.75
                                        )
                          )




pred_test$pred_bst <- predict(gradient_boost, test_x)

error <- error %>%
  bind_rows(c(modelo = "bst", error = rmse(pred_test$price, pred_test$pred_bst)))
```

De los modelos que hemos entrenado, **¿con cuál te quedarías y por qué?**

Explicatividad

Esribe y comenta los gráficos que consideres necesarios para interpretar el algoritmo de *Gradient boosting*.