

Creación de modelo lineal

Cargamos las librerías a utilizar

```
library(tidymodels) # Nos permitira crear el algoritmo de regresión lineal.
library(dplyr) # Manipulación de datos.
library(ggplot2) # Gráficar.
library(caret) # Usaremos para las métricas de desempeño.
```

Cargamos la base de datos limpia.

```
df<-read.csv("houses_to_rent_clear.csv",sep=";")
```

Preprocesado de los datos.

```
df<- df %>%
  mutate(accept=ifelse(accept=="accept",1,0),
         furnished=ifelse(furnished=="furnished",1,0))
# Transformamos las variables cetegoricas a numérica.
```

```
df_scale<- df %>%
  mutate(rooms=as.numeric(scale(rooms)),
         bathroom=as.numeric(scale(bathroom)),
         parking.spaces=as.numeric(scale(parking.spaces)),
         floor=as.numeric(scale(floor))
  )
# Realizamos un rescalado de los datos. Para que pueda ser comparables entre si.
```

Dividimos los datos de entrenamiento y validación.

```
set.seed(2018)

split<-initial_split(df_scale,prop = 0.8,
                     strata = rent.amount) # Dividimos los datos un 80% para entrenaar y el resto para

training_data<-split %>%
  training() # Seleccionamos los datos de entrenamiento.

test_data<-split %>%
  testing() # Seleccionamos los datos de validación.
```

Creación del modelo de regresión lineal.

```
lm_model<-linear_reg() %>%  
  fit(rent.amount~.,data=training_data)
```

```
test_results <- predict(lm_model, new_data = test_data) %>%  
  bind_cols(test_data)
```

R^2

Es una métrica de desempeño. Que mide el grado de covarianza entre el valor predicho y el valor actual. Entre más cercano sea a 1 mayor será la cercanía a los valores originales.

```
R2(predict(lm_model,training_data),training_data$rent.amount)
```

```
##           .pred  
## [1,] 0.9358208
```

```
R2(test_results$.pred,test_results$rent.amount)
```

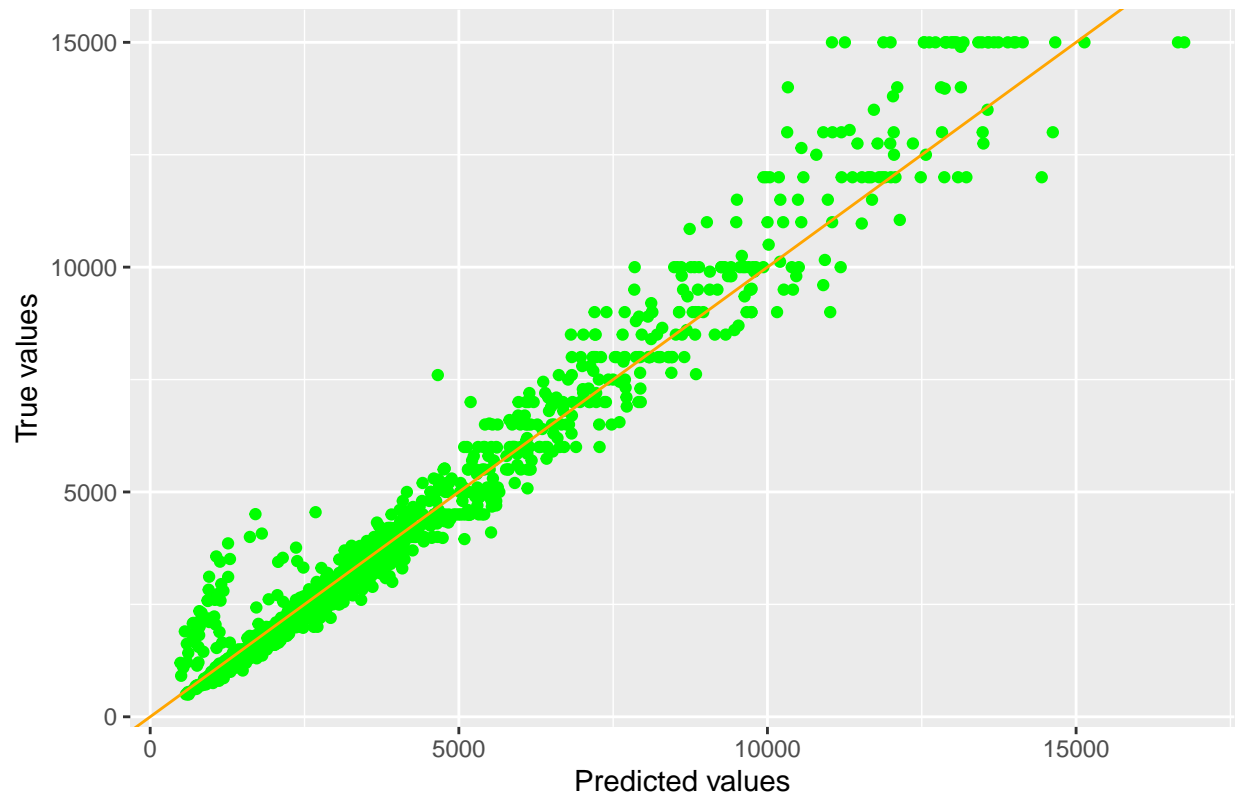
```
## [1] 0.9217225
```

Observamos el R^2 bastante alto. Tanto para los datos de entrenamiento y validación. Por cual puede explicar el modelo la mayoría de los datos. Lo cual generalizar bien.

Representación gráfica.

```
test_results %>%  
  mutate(.pred=exp(.pred),rent.amount=exp(rent.amount)) %>%  
  ggplot(aes(x=.pred,y=rent.amount)) +  
  geom_point(color="green") +  
  
  geom_abline(intercept = 0, slope = 1, color = 'orange') +  
  labs(title = "Linear Regression Results") +  
  xlab("Predicted values") +  
  ylab("True values")
```

Linear Regression Results



Guardamos el modelo.

```
saveRDS(lm_model , "lm_model_houses.rds " )
```