

# Proyecto Final1.0

Carlos Alvarado Ronald Guerra Rene Hernandez

r Sys.Date()

```
dataset = read.csv("train.csv")
```

```
#dataset
```

```
summary(dataset)
```

```
##          id          longitude          latitude housing_median_age
## Min.      :    1    Min.      :-124.3    Min.      :32.54    Min.      : 1.00
## 1st Qu.: 5140    1st Qu.: -121.8    1st Qu.:33.93    1st Qu.:18.00
## Median :10210    Median : -118.5    Median :34.26    Median :29.00
## Mean     :10275    Mean     :-119.6    Mean     :35.64    Mean     :28.85
## 3rd Qu.:15449    3rd Qu.: -118.0    3rd Qu.:37.72    3rd Qu.:37.00
## Max.     :20640    Max.     :-114.3    Max.     :41.95    Max.     :52.00
##
## total_rooms total_bedrooms population households
## Min.      :    2    Min.      : 1.0    Min.      :    6    Min.      : 1.0
## 1st Qu.: 1444    1st Qu.: 295.0    1st Qu.:   786    1st Qu.: 280.0
## Median : 2121    Median : 433.0    Median : 1163    Median : 408.0
## Mean     : 2635    Mean     : 537.8    Mean     : 1425    Mean     : 500.1
## 3rd Qu.: 3138    3rd Qu.: 647.0    3rd Qu.: 1722    3rd Qu.: 604.5
## Max.     :39320    Max.     :6445.0    Max.     :28566    Max.     :6082.0
##
## NA's      :137
## median_income median_house_value ocean_proximity
## Min.      : 0.4999    Min.      : 14999    Length:14447
## 1st Qu.: 2.5671    1st Qu.:119600    Class :character
## Median : 3.5350    Median :179700    Mode  :character
## Mean     : 3.8639    Mean     :206874
## 3rd Qu.: 4.7229    3rd Qu.:264600
## Max.     :15.0001    Max.     :500001
##
```

id: Identificador único para cada propiedad. longitude: Longitud geográfica de la propiedad. latitude: Latitud geográfica de la propiedad. housing\_median\_age: Edad media de la vivienda en años. total\_rooms: Número total de habitaciones. total\_bedrooms: Número total de dormitorios. population: Población en el área de la propiedad. households: Número de hogares en el área de la propiedad. median\_income: Ingreso medio de los hogares en el área de la propiedad. median\_house\_value: Valor medio de las viviendas en el área de la propiedad (esta podría ser una variable objetivo para un modelo de regresión). ocean\_proximity: Proximidad al océano (parece ser una variable categórica).

```
# Calcula el número de NA en cada columna
na_count <- colSums(is.na(dataset))
```

```
# Muestra las columnas que tienen al menos un NA
na_columns <- names(dataset)[na_count > 0]
```

```
print(na_columns)
```

```
## [1] "total_bedrooms"
```

**\*\* total\_bedrooms contiene NA's que deberan ser imputados en la fase de preparacion de datos \*\***

```
str(dataset)
```

```
## 'data.frame': 14447 obs. of 11 variables:
## $ id : int 9744 13893 18277 16176 8843 7653 14056 18819 17145 16187 ...
## $ longitude : num -122 -116 -122 -122 -118 ...
## $ latitude : num 36.8 34.1 37.3 37.7 34.1 ...
## $ housing_median_age: int 15 37 35 52 28 28 23 40 18 52 ...
## $ total_rooms : int 2191 452 1172 126 4001 2152 3999 690 1636 107 ...
## $ total_bedrooms : int 358 109 184 24 1352 415 1182 129 414 79 ...
## $ population : int 1150 184 512 37 1799 1623 2051 305 853 167 ...
## $ households : int 330 59 175 27 1220 429 1130 110 439 53 ...
## $ median_income : num 4.8 3.73 7.36 10.23 2.58 ...
## $ median_house_value: num 227500 65800 500001 225000 272900 ...
## $ ocean_proximity : chr "<1H OCEAN" "INLAND" "<1H OCEAN" "NEAR BAY" ...
```

Rellenar datos los 137 datos de la columna “total\_bedrooms” usando k-means:

```
# Cargando la librería necesaria
```

```
#install.packages("mice")
```

```
library(mice)
```

```
## Warning: package 'mice' was built under R version 4.2.3
```

```
##
```

```
## Attaching package: 'mice'
```

```
## The following object is masked from 'package:stats':
```

```
##
```

```
## filter
```

```
## The following objects are masked from 'package:base':
```

```
##
```

```
## cbind, rbind
```

```
# Imputación por K-Nearest Neighbors (KNN)
```

```
tempData <- mice(dataset, method='pmm', m=5) # puedes cambiar el valor de m si es necesario
```

```
##
## iter imp variable
## 1 1 total_bedrooms
## 1 2 total_bedrooms
## 1 3 total_bedrooms
## 1 4 total_bedrooms
## 1 5 total_bedrooms
## 2 1 total_bedrooms
## 2 2 total_bedrooms
## 2 3 total_bedrooms
## 2 4 total_bedrooms
## 2 5 total_bedrooms
## 3 1 total_bedrooms
## 3 2 total_bedrooms
## 3 3 total_bedrooms
## 3 4 total_bedrooms
## 3 5 total_bedrooms
## 4 1 total_bedrooms
## 4 2 total_bedrooms
## 4 3 total_bedrooms
## 4 4 total_bedrooms
## 4 5 total_bedrooms
## 5 1 total_bedrooms
## 5 2 total_bedrooms
## 5 3 total_bedrooms
## 5 4 total_bedrooms
## 5 5 total_bedrooms
```

```
## Warning: Number of logged events: 1
```

```
dataset <- complete(tempData)
```

```
summary(dataset)
```

```
##          id          longitude          latitude housing_median_age
## Min.      :    1   Min.      :-124.3   Min.      :32.54   Min.      : 1.00
## 1st Qu.: 5140   1st Qu.: -121.8   1st Qu.: 33.93   1st Qu.: 18.00
## Median :10210   Median : -118.5   Median : 34.26   Median : 29.00
## Mean    :10275   Mean    : -119.6   Mean     : 35.64   Mean     : 28.85
## 3rd Qu.:15449   3rd Qu.: -118.0   3rd Qu.: 37.72   3rd Qu.: 37.00
## Max.     :20640   Max.     : -114.3   Max.     : 41.95   Max.     : 52.00
## total_rooms total_bedrooms population households
## Min.      :    2   Min.      : 1.0   Min.      :    6   Min.      : 1.0
## 1st Qu.: 1444   1st Qu.: 295.0   1st Qu.:   786   1st Qu.: 280.0
## Median : 2121   Median : 434.0   Median : 1163   Median : 408.0
## Mean     : 2635   Mean     : 538.1   Mean      : 1425   Mean      : 500.1
## 3rd Qu.: 3138   3rd Qu.: 647.0   3rd Qu.: 1722   3rd Qu.: 604.5
## Max.     :39320   Max.     :6445.0   Max.     :28566   Max.     :6082.0
## median_income median_house_value ocean_proximity
## Min.      : 0.4999   Min.      : 14999   Length:14447
## 1st Qu.: 2.5671   1st Qu.:119600   Class :character
## Median : 3.5350   Median :179700   Mode  :character
## Mean      : 3.8639   Mean      :206874
```

```
## 3rd Qu.: 4.7229 3rd Qu.:264600
## Max. :15.0001 Max. :500001
```

\*\* Datos Anteriores \*\*

```
id longitude latitude housing_median_age total_rooms total_bedrooms population
Min. : 1 Min. :-124.3 Min. :32.54 Min. : 1.00 Min. : 2 Min. : 1.0 Min. : 6
1st Qu.: 5140 1st Qu.: -121.8 1st Qu.:33.93 1st Qu.:18.00 1st Qu.: 1444 1st Qu.: 295.0 1st Qu.: 786
Median :10210 Median :-118.5 Median :34.26 Median :29.00 Median : 2121 Median : 433.0 Median : 1163
Mean :10275 Mean :-119.6 Mean :35.64 Mean :28.85 Mean : 2635 Mean : 537.8 Mean : 1425
3rd Qu.:15449 3rd Qu.: -118.0 3rd Qu.:37.72 3rd Qu.:37.00 3rd Qu.: 3138 3rd Qu.: 647.0 3rd Qu.: 1722
Max. :20640 Max. :-114.3 Max. :41.95 Max. :52.00 Max. :39320 Max. :6445.0 Max. :28566
NA's :137
households median_income median_house_value ocean_proximity
Min. : 1.0 Min. : 0.4999 Min. : 14999 Length:14447
1st Qu.: 280.0 1st Qu.: 2.5671 1st Qu.:119600 Class :character
Median : 408.0 Median : 3.5350 Median :179700 Mode :character
Mean : 500.1 Mean : 3.8639 Mean :206874
3rd Qu.: 604.5 3rd Qu.: 4.7229 3rd Qu.:264600
Max. :6082.0 Max. :15.0001 Max. :500001
```

```
str(dataset)
```

```
## 'data.frame': 14447 obs. of 11 variables:
## $ id : int 9744 13893 18277 16176 8843 7653 14056 18819 17145 16187 ...
## $ longitude : num -122 -116 -122 -122 -118 ...
## $ latitude : num 36.8 34.1 37.3 37.7 34.1 ...
## $ housing_median_age: int 15 37 35 52 28 28 23 40 18 52 ...
## $ total_rooms : int 2191 452 1172 126 4001 2152 3999 690 1636 107 ...
## $ total_bedrooms : int 358 109 184 24 1352 415 1182 129 414 79 ...
## $ population : int 1150 184 512 37 1799 1623 2051 305 853 167 ...
## $ households : int 330 59 175 27 1220 429 1130 110 439 53 ...
## $ median_income : num 4.8 3.73 7.36 10.23 2.58 ...
## $ median_house_value: num 227500 65800 500001 225000 272900 ...
## $ ocean_proximity : chr "<1H OCEAN" "INLAND" "<1H OCEAN" "NEAR BAY" ...
```

```
** Datos anteriores ** 'data.frame': 14447 obs. of 11 variables: $ id : int 9744 13893 18277 16176 8843 7653
14056 18819 17145 16187 ... $ longitude : num -122 -116 -122 -122 -118 ... $ latitude : num 36.8 34.1 37.3
37.7 34.1 ... $ housing_median_age: int 15 37 35 52 28 28 23 40 18 52 ... $ total_rooms : int 2191 452
1172 126 4001 2152 3999 690 1636 107 ... $ total_bedrooms : int 358 109 184 24 1352 415 1182 129 414
79 ... $ population : int 1150 184 512 37 1799 1623 2051 305 853 167 ... $ households : int 330 59 175 27
1220 429 1130 110 439 53 ... $ median_income : num 4.8 3.73 7.36 10.23 2.58 ... $ median_house_value:
num 227500 65800 500001 225000 272900 ... $ ocean_proximity : chr "<1H OCEAN" "INLAND" "<1H
OCEAN" "NEAR BAY" ...
```

```
** Los posibles resultados de ocean_proximity estan limitados a 5 opciones: "<1H OCEAN" "INLAND"
"NEAR BAY" "NEAR OCEAN" "ISLAND" **
```

```
# Selecciona solo las columnas numéricas
numeric_columns <- dataset[sapply(dataset, is.numeric)]

# Calcula la matriz de correlaciones para las columnas numéricas
cor_matrix <- cor(numeric_columns, use = "complete.obs")
cor_matrix
```

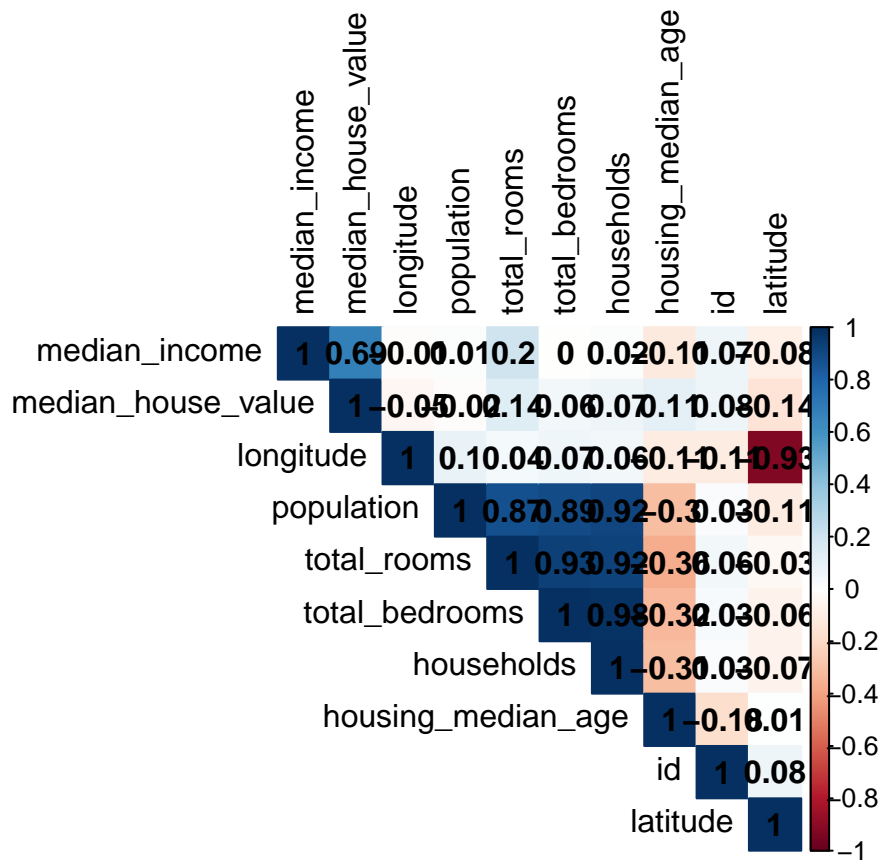
```
##          id    longitude    latitude housing_median_age
## id          1.00000000 -0.10989609  0.077493954      -0.176502064
## longitude   -0.10989609  1.00000000 -0.925404909      -0.108182783
## latitude    0.07749395 -0.92540491  1.000000000        0.009715456
## housing_median_age -0.17650206 -0.10818278  0.009715456        1.000000000
## total_rooms   0.05914449  0.04279013 -0.033945813      -0.360963978
## total_bedrooms 0.03059012  0.06679148 -0.064201301      -0.322477881
## population    0.02523310  0.09985011 -0.106931755      -0.303717353
## households    0.02881578  0.05515997 -0.069274646      -0.305342533
## median_income  0.07221005 -0.01325143 -0.081381915      -0.114156281
## median_house_value 0.07772461 -0.04888187 -0.143545253        0.111424346
##          total_rooms total_bedrooms population households
## id          0.05914449    0.030590122  0.02523310  0.02881578
## longitude    0.04279013    0.066791477  0.09985011  0.05515997
## latitude     -0.03394581   -0.064201301 -0.10693175 -0.06927465
## housing_median_age -0.36096398   -0.322477881 -0.30371735 -0.30534253
## total_rooms   1.00000000    0.930319648  0.87019710  0.92055490
## total_bedrooms 0.93031965    1.000000000  0.89003447  0.98178431
## population    0.87019710    0.890034468  1.00000000  0.91591932
## households    0.92055490    0.981784309  0.91591932  1.00000000
## median_income  0.20199254   -0.001102666  0.01163270  0.01834605
## median_house_value 0.13755605    0.057212574 -0.01921386  0.07138869
##          median_income median_house_value
## id          0.072210048      0.07772461
## longitude    -0.013251433      -0.04888187
## latitude     -0.081381915      -0.14354525
## housing_median_age -0.114156281      0.11142435
## total_rooms    0.201992543      0.13755605
## total_bedrooms -0.001102666      0.05721257
## population     0.011632696     -0.01921386
## households     0.018346055      0.07138869
## median_income  1.000000000      0.68720046
## median_house_value 0.687200464      1.00000000
```

```
library(corrplot)
```

```
## Warning: package 'corrplot' was built under R version 4.2.3
```

```
## corrplot 0.92 loaded
```

```
# Grafica la matriz de correlaciones
corrplot(cor_matrix, method = "color", type = "upper", order = "hclust",
  addCoef.col = "black", # Add correlation coefficient on the plot
  tl.col="black", # Text label color
  #tl.srt=45
) # Text label rotation
```



```
#install.packages("caret")
library(caret)
```

```
## Warning: package 'caret' was built under R version 4.2.3
```

```
## Loading required package: ggplot2
```

```
## Warning: package 'ggplot2' was built under R version 4.2.3
```

```
## Loading required package: lattice
```

Crear una partición de los datos: Divide tu conjunto de datos en un conjunto de entrenamiento y un conjunto de pruebas. Esto es esencial para evaluar el rendimiento del modelo en datos no vistos.

```
set.seed(123) # Para reproducibilidad
index <- createDataPartition(dataset$median_house_value, p = 0.8, list = FALSE)
training_data <- dataset[index, ]
testing_data <- dataset[-index, ]
```

```
summary(training_data)
```

```
##      id      longitude      latitude      housing_median_age
## Min.   :      1   Min.   : -124.3   Min.   : 32.54   Min.   : 1.00
```

```
## 1st Qu.: 5100 1st Qu.: -121.8 1st Qu.: 33.94 1st Qu.: 18.00
## Median :10232 Median : -118.5 Median : 34.27 Median : 29.00
## Mean :10278 Mean : -119.6 Mean : 35.66 Mean : 28.85
## 3rd Qu.:15482 3rd Qu.: -118.0 3rd Qu.: 37.72 3rd Qu.: 37.00
## Max. :20640 Max. : -114.3 Max. : 41.95 Max. : 52.00
## total_rooms total_bedrooms population households
## Min. : 8 Min. : 1.0 Min. : 8 Min. : 1.0
## 1st Qu.: 1452 1st Qu.: 296.0 1st Qu.: 786 1st Qu.: 279.0
## Median : 2126 Median : 435.0 Median : 1168 Median : 410.0
## Mean : 2637 Mean : 540.6 Mean : 1433 Mean : 502.4
## 3rd Qu.: 3149 3rd Qu.: 652.0 3rd Qu.: 1727 3rd Qu.: 608.0
## Max. :39320 Max. :6445.0 Max. :28566 Max. :6082.0
## median_income median_house_value ocean_proximity
## Min. : 0.4999 Min. : 14999 Length:11560
## 1st Qu.: 2.5625 1st Qu.:119600 Class :character
## Median : 3.5278 Median :179750 Mode :character
## Mean : 3.8512 Mean :206688
## 3rd Qu.: 4.7143 3rd Qu.:264600
## Max. :15.0001 Max. :500001
```

```
summary(testing_data)
```

```
## id longitude latitude housing_median_age
## Min. : 3 Min. : -124.3 Min. : 32.56 Min. : 1.00
## 1st Qu.: 5248 1st Qu.: -121.7 1st Qu.: 33.91 1st Qu.: 18.00
## Median :10155 Median : -118.5 Median : 34.21 Median : 29.00
## Mean :10267 Mean : -119.5 Mean : 35.56 Mean : 28.84
## 3rd Qu.:15315 3rd Qu.: -118.0 3rd Qu.: 37.71 3rd Qu.: 37.50
## Max. :20639 Max. : -114.7 Max. : 41.86 Max. : 52.00
## total_rooms total_bedrooms population households
## Min. : 2 Min. : 2.0 Min. : 6.0 Min. : 2
## 1st Qu.: 1424 1st Qu.: 293.0 1st Qu.: 786.5 1st Qu.: 282
## Median : 2107 Median : 428.0 Median : 1145.0 Median : 403
## Mean : 2629 Mean : 527.9 Mean : 1395.3 Mean : 491
## 3rd Qu.: 3085 3rd Qu.: 631.0 3rd Qu.: 1696.0 3rd Qu.: 587
## Max. :30450 Max. :5033.0 Max. :13251.0 Max. :4339
## median_income median_house_value ocean_proximity
## Min. : 0.4999 Min. : 14999 Length:2887
## 1st Qu.: 2.5882 1st Qu.:119500 Class :character
## Median : 3.5750 Median :179700 Mode :character
## Mean : 3.9148 Mean :207617
## 3rd Qu.: 4.7468 3rd Qu.:264450
## Max. :15.0001 Max. :500001
```

## Primer Modelo

```
# Entrenar el modelo de regresión lineal
```

```
modell1 <- train(median_house_value ~ ., data = training_data, method = "lm", trControl = trainControl(m
```

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

```

# Hacer la predicción con la parte de prueba
predictions <- predict(model1, testing_data)

# Calcular el Error Cuadrático Medio
mse <- mean((predictions - testing_data$median_house_value)^2)

# Calcular la Raíz del Error Cuadrático Medio
rmse <- sqrt(mse)

# Calcular la desviación estándar de la variable median_house_value en el conjunto de datos de prueba
std_dev <- sd(testing_data$median_house_value)

# Mostrar la Raíz del Error Cuadrático Medio
print(paste("Raíz del Error Cuadrático Medio (RMSE):", rmse))

```

```
## [1] "Raíz del Error Cuadrático Medio (RMSE): 68430.7188148631"
```

```

# Mostrar la desviación estándar de la variable median_house_value
print(paste("Desviación Estándar de median_house_value:", std_dev))

```

```
## [1] "Desviación Estándar de median_house_value: 116005.531136358"
```

## Segundo Modelo

```

model2 <- train(median_house_value ~ ., data = training_data, method = "glmnet",
                trControl = trainControl(method = "cv", number = 10),
                tuneGrid = expand.grid(alpha = 1, lambda = seq(0.001, 0.1, length = 10)))

# Hacer la predicción con la parte de prueba
predictions <- predict(model2, testing_data)

# Calcular el Error Cuadrático Medio
mse <- mean((predictions - testing_data$median_house_value)^2)

# Calcular la Raíz del Error Cuadrático Medio
rmse <- sqrt(mse)

# Calcular la desviación estándar de la variable median_house_value en el conjunto de datos de prueba
std_dev <- sd(testing_data$median_house_value)

# Mostrar la Raíz del Error Cuadrático Medio
print(paste("Raíz del Error Cuadrático Medio (RMSE):", rmse))

```

```
## [1] "Raíz del Error Cuadrático Medio (RMSE): 68462.5578208677"
```

```

# Mostrar la desviación estándar de la variable median_house_value
print(paste("Desviación Estándar de median_house_value:", std_dev))

```

```
## [1] "Desviación Estándar de median_house_value: 116005.531136358"
```



## Tercer modelo

```
# Realizar la normalización de las variables numéricas en el conjunto de datos
normalize <- function(x) {
  return((x - min(x)) / (max(x) - min(x)))
}

# Aplicar la normalización a las variables numéricas en el conjunto de datos
numeric_cols <- sapply(training_data, is.numeric)
training_data_normalized <- training_data
training_data_normalized[numeric_cols] <- lapply(training_data_normalized[numeric_cols], normalize)

# Calcular la desviación estándar de la variable objetivo en el conjunto de datos de prueba
std_dev <- sd(testing_data$median_house_value)

# Entrenar el modelo de regresión lineal con los datos normalizados
model3 <- train(median_house_value ~ ., data = training_data_normalized, method = "lm", trControl = tra

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

# Normalizar las variables numéricas en el conjunto de prueba
testing_data_normalized <- testing_data
testing_data_normalized[numeric_cols] <- lapply(testing_data_normalized[numeric_cols], normalize)

# Hacer la predicción con el conjunto de prueba normalizado
predictions <- predict(model3, testing_data_normalized)

# Calcular el Error Cuadrático Medio con los datos normalizados
mse <- mean((predictions - testing_data_normalized$median_house_value)^2)

# Calcular la Raíz del Error Cuadrático Medio con los datos normalizados
rmse <- sqrt(mse)

std_dev <- sd(testing_data$median_house_value)

# Mostrar la Raíz del Error Cuadrático Medio con los datos normalizados
print(paste("Raiz del Error Cuadratico Medio (RMSE) con datos normalizados:", rmse))

## [1] "Raiz del Error Cuadratico Medio (RMSE) con datos normalizados: 0.186640861280498"

# Mostrar la desviación estándar de la variable median_house_value
print(paste("Desviacion Estandar de median_house_value:", std_dev))

## [1] "Desviacion Estandar de median_house_value: 116005.531136358"
```