

Examen Argumentativo

Emmanuel Naranjo Blanco – A00835704

Tecnológico de Monterrey – Campus Monterrey

Fecha de entrega: 01/12/2023

Profesor: Dr. Alexander Garrido Ríos

I. Introducción

En este reporte argumentativo se presentan los modelos de regresión lineal con fines de predicción, cuyo objetivo es

evaluar el rendimiento del modelo en un conjunto de validación y utilizar las métricas de predicción.

A continuación, se describirán los algoritmos para la selección de variables independientes que se aplicarán en el procedimiento de regresión lineal múltiple (MLR).

Inicialmente se realizará el análisis EDA, con el objetivo de examinar y comprender la estructura, patrones y características fundamentales de un conjunto de datos.

Seguidamente se determinará si es necesario o no el proceso de imputación de datos con el objetivo de preparar los datos para el análisis MLR y que se obtenga la mejor estimación posible.

Por último, se empleará la etapa de modelado MLR. En otras palabras utilizaremos un dataset cuyo nombre es “city” de 50 instancias para 7 atributos de tipo numérico “X1”, “X2”, “X3”, “X4”, “X5”, “X6” y “X7”; cuya descripción se muestra a continuación:

- X1 = ‘total overall reported crime rate per 1 million residents
- X2 = “reported violent crime rate per 100,000 residents”
- X3 = “annual police funding in S/resident”
- X4 = “% of people 25 yearst with 4 yrs. of high school”
- X5 = “% of 16-to-19-year-old not in high school and not high school graduates”
- X6 = “% of 18-24-year-old in college”
- X7 = “% of people 25 years+ with at least 4 years of college”

II. Exploración de Datos

i. Importación de Librerías y Dataset

Inicialmente se instalan los siguientes paquetes y se procede con cargar la base de datos.

```
library("statsr")  
## Warning: package 'statsr' was built under R version 4.3.2  
## Loading required package: BayesFactor  
## Warning: package 'BayesFactor' was built under R version 4.3.2  
## Loading required package: coda  
## Warning: package 'coda' was built under R version 4.3.2  
## Loading required package: Matrix  
## Warning: package 'Matrix' was built under R version 4.3.2  
## *****  
## Welcome to BayesFactor 0.9.12-4.5. If you have questions, please contact R  
ichard Morey (richarddmorey@gmail.com).
```

```
##
## Type BFManual() to open the manual.
## *****

  library("skimr")
  ## Warning: package 'skimr' was built under R version 4.3.2
  library("forecast")
  ## Warning: package 'forecast' was built under R version 4.3.2
  ## Registered S3 method overwritten by 'quantmod':
## method          from
## as.zoo.data.frame zoo
  library("ggplot2")
  ## Warning: package 'ggplot2' was built under R version 4.3.2
  library("dplyr")
  ## Warning: package 'dplyr' was built under R version 4.3.2
  ##
## Attaching package: 'dplyr'
  ## The following objects are masked from 'package:stats':
##
##   filter, lag
  ## The following objects are masked from 'package:base':
##
##   intersect, setdiff, setequal, union
  library("broom")
  ## Warning: package 'broom' was built under R version 4.3.2
  library("ggpubr")
  ## Warning: package 'ggpubr' was built under R version 4.3.2
  ##
## Attaching package: 'ggpubr'
  ## The following object is masked from 'package:forecast':
##
##   gghistogram
  library("gvlma")
  ## Warning: package 'gvlma' was built under R version 4.3.1
```

```

library("readxl")
## Warning: package 'readxl' was built under R version 4.3.2
library("caret")
## Warning: package 'caret' was built under R version 4.3.2
## Loading required package: lattice
library("tidyverse")
## Warning: package 'tidyverse' was built under R version 4.3.2
## Warning: package 'tidyr' was built under R version 4.3.2
## Warning: package 'readr' was built under R version 4.3.2
## Warning: package 'purrr' was built under R version 4.3.2
## Warning: package 'forcats' was built under R version 4.3.2
## Warning: package 'lubridate' was built under R version 4.3.2
## — Attaching core tidyverse packages ————— tidyverse
rse 2.0.0 —
## ✓ forcats   1.0.0   ✓ stringr   1.5.0
## ✓ lubridate 1.9.3   ✓ tibble    3.2.1
## ✓ purrr     1.0.2   ✓ tidyr     1.3.0
## ✓ readr     2.1.4
## — Conflicts ————— tidyverse_co
nflicts() —
## ✗ tidyr::expand() masks Matrix::expand()
## ✗ dplyr::filter() masks stats::filter()
## ✗ dplyr::lag()   masks stats::lag()
## ✗ purrr::lift()  masks caret::lift()
## ✗ tidyr::pack()  masks Matrix::pack()
## ✗ tidyr::unpack() masks Matrix::unpack()
## ⓘ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all
conflicts to become errors
library("pillar")
## Warning: package 'pillar' was built under R version 4.3.2
##
## Attaching package: 'pillar'

```

```
##
## The following object is masked from 'package:dplyr':
##
##      dim_desc
      library("psych")
      ## Warning: package 'psych' was built under R version 4.3.2
      ##
## Attaching package: 'psych'
##
## The following objects are masked from 'package:ggplot2':
##
##      %+%, alpha
      library("readr")
library("GGally")
      ## Warning: package 'GGally' was built under R version 4.3.2
      ## Registered S3 method overwritten by 'GGally':
## method from
## +.gg      ggplot2
      library("corrplot")
      ## Warning: package 'corrplot' was built under R version 4.3.2
      ## corrplot 0.92 loaded
      library("reshape2")
      ## Warning: package 'reshape2' was built under R version 4.3.2
      ##
## Attaching package: 'reshape2'
##
## The following object is masked from 'package:tidyr':
##
##      smiths
      library("gmodels")
      ## Warning: package 'gmodels' was built under R version 4.3.2
      library("mice")
      ## Warning: package 'mice' was built under R version 4.3.2
```

```
##
## Attaching package: 'mice'
##
## The following object is masked from 'package:pillar':
##
##     squeeze
##
## The following object is masked from 'package:stats':
##
##     filter
##
## The following objects are masked from 'package:base':
##
##     cbind, rbind
data <- read.csv("city.csv", sep=";")
View(data)
```

ii. Visualización de los Atributos y sus Datos

Iniciaremos con la exploración de datos inicial, la cual permitirá revisar los tipos atributos, la existencia o no de missing data, los percentiles, la distribución de los datos; entre otros.

La función `dim` proporciona las dimensiones del dataframe, es decir, el número de instancias y atributos. Este es un paso básico pero crucial para entender la dimensión del conjunto de datos. De la cual se obtiene que estamos trabajando con una estructura de 50 instancias y 7 atributos.

```
dim(data)
## [1] 50  7
```

La función `skim` proporciona rápidamente una visión general del marco de los datos. En este caso, nos muestra que el número de valores que faltan es cero, así como las características estadísticas básicas.

Es por esta razón que para efectos del presente análisis no se empleará imputación de datos.

```
skim(data)
Data summary

Name                                data
Number of rows                      50
Number of columns                    7








_____

Column type frequency:
numeric                             7

_____

Group variables                      None
```

Variable type: numeric

	s	n	c									
kim_vari	_missi	omplete										
able	ng	_rate	ean	d	0	25	50	75	100	ist		
1	X	0	1	17.96	93.94	41	97.00	54.5	20.50	740		—
2	X	0	1	16.18	73.74	9	30.75	54.0	22.50	545		—
3	X	0	1	7.76	3.82	6	0.00	4.5	2.25	6		—
4	X	0	1	8.80	.97	2	9.00	9.0	7.00	1		—
5	X	0	1	5.40	.02		1.00	4.0	9.00	4		—
6	X	0	1	9.90	4.80		1.25	5.0	4.25	1		—
7	X	0	1	3.82	.16		1.00	2.0	5.75	6		—

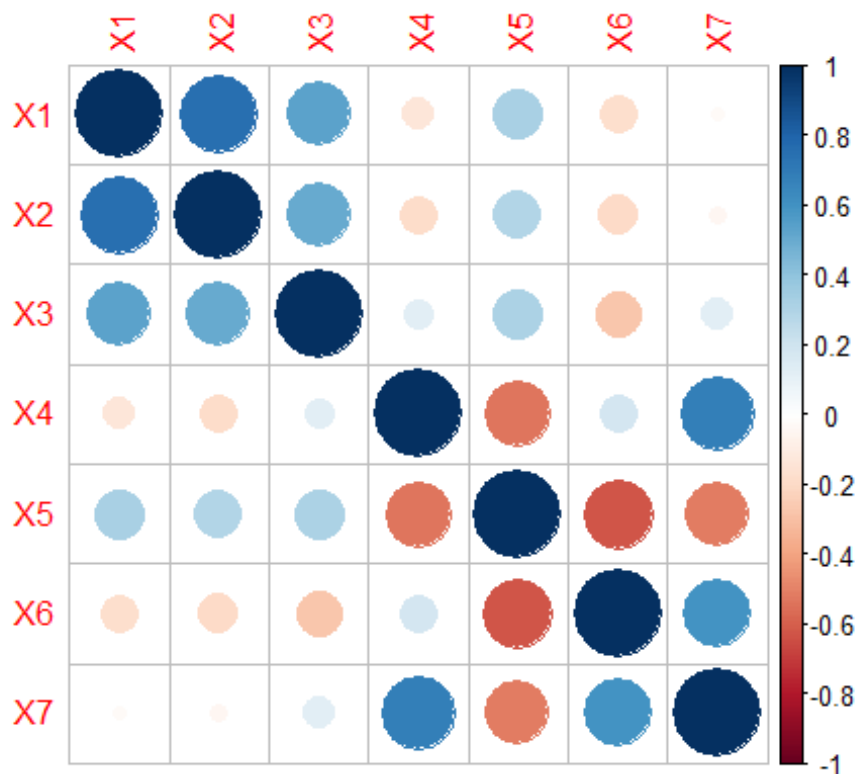
Una representación de cómo los datos interactúan entre sí es mediante la matriz de correlación.

Esto permitirá evaluar las relaciones lineales, analizar la multicolinealidad y seleccionar los predictores para la variable de salida.

Visualmente esto es, entre mayor el diámetro del círculo mostrado en la gráfica, mayor es la correlación entre las variables. O bien, en la tabla de correlación: 1 indica una correlación positiva perfecta, -1 indica una correlación negativa perfecta, 0 indica falta de correlación lineal.

```
cor.matrix=cor(data)  
corrplot(cor.matrix)
```

```
view(cor.matrix)
```



Como resultados se tiene lo siguiente:

Fuerte correlación positiva:

- X1 y X2 • X4 y X7

Fuerte correlación negativa:

- X4 y X5 • X5 con X4 y X6

Multicolinealidad:

• Algunas variables, como X4 y X7, tienen una alta correlación, lo que podría indicar multicolinealidad. En la práctica, esto podría afectar la interpretación de un modelo de regresión.

What factors influence the total overall reporte crime rate per 1 million residents?

- Para responder a esta pregunta, examinaremos las correlaciones entre la variable de interés (X1) con respecto a las demás variables explicativas (X2 a X7):
- La correlación más fuerte positiva con X1 es con X2: “reported violent crime rate per 100,000 residents” con un valor de 0.76. Esto sugiere que a medida que la tasa de criminalidad violenta reportada aumenta, la tasa total de criminalidad también tiende a aumentar.
- La variable X3 “annual police funding in \$/resident” muestra una correlación positiva moderada con X1 de 0.53. Esto indicaría que a medida que aumenta el financiamiento policial por residente, la tasa total de criminalidad tiende a aumentar.
- La correlación positiva moderada de X1 con X5: “% of 16-to-19-year-old not in high school and not high school graduates”, sugiere que entre el porcentaje de

personas de 16 a 19 años que no están en la escuela y no han completado la escuela secundaria aumenta, la tasa total general de criminalidad aumenta.

- Además, X4: “% of people 25 years with 4 years of high school”, X6: “% of 18-24-year-old in college”, y X7: “% of people 25 years+ with at least 4 years of college”, tienen correlaciones con X1 negativas (aunque casi insignificativa), lo que sugiere una relación inversa entre la tasa de criminalidad y el número de personas con niveles de educación.

De este modo, basándonos en las correlaciones observadas con este dataset, podemos concluir que los factores X2, X3, X5 pueden influir proporcionalmente en X1.

III. Construcción del Modelo de Regresión Lineal

Múltiple

Una vez analizado qué miden los distintos predictores y por qué son relevantes para predecir la variable de salida, procederemos con la propuesta de una ecuación de regresión lineal múltiple para predecir el valor de X_1 .

Definimos la hipótesis de regresión lineal múltiple para la variable dependiente $Y=X_1$ de la siguiente manera:

$$Y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_p x_p + \epsilon$$

donde β_0, \dots, β_p son coeficientes y ϵ es el ruido o parte no explicada.

A continuación, se presenta la ecuación MLR para X_1 con su respectivo análisis estadístico.

i. X_1 en función de todas las variables predictoras

En un primer acercamiento a encontrar nuestro mejor MLR, seleccionaremos todas las variables predictoras.

```
modelo_lineal_1 <- lm(X1 ~ ., data = data)

# Resumen del modelo
options(scipen = 999)
summary(modelo_lineal_1)

##
## Call:
## lm(formula = X1 ~ ., data = data)
##
```

```
## Residuals:
##      Min       1Q   Median       3Q      Max
## -291.82 -105.31  -26.78   85.62  705.89
##
## Coefficients:
##              Estimate Std. Error t value    Pr(>|t|)
## (Intercept) 100.39361   370.69317   0.271     0.788
## X2           0.33234    0.05962   5.574 0.00000152 ***
## X3           3.99817    2.68248   1.490     0.143
## X4           1.85791    5.24087   0.355     0.725
## X5           7.83886    7.75987   1.010     0.318
## X6           2.55877    3.42695   0.747     0.459
## X7          -3.23116   10.71537  -0.302     0.764
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 195.2 on 43 degrees of freedom
## Multiple R-squared:  0.6132, Adjusted R-squared:  0.5592
## F-statistic: 11.36 on 6 and 43 DF,  p-value: 0.0000001424
```

ii. X1 en función de las variables predictoras X2, X3 y X5

En un segundo acercamiento a encontrar nuestro mejor MLR, seleccionaremos las variables X2, X3 y X5, basándonos en las correlaciones observadas anteriormente, ya que pueden proporcionar información valiosa para predecir la variable X1 en un modelo de regresión lineal múltiple.

```
modelo_lineal_2 <- lm(X1 ~ X2 + X3 + X5, data = data)

# Resumen del modelo
options(scipen = 999)
summary(modelo_lineal_2)
```

```
##
## Call:
## lm(formula = X1 ~ X2 + X3 + X5, data = data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -316.94 -108.84  -31.95   81.89  681.29
##
## Coefficients:
##              Estimate Std. Error t value    Pr(>|t|)
## (Intercept)  309.80151   94.51353   3.278    0.002 **
## X2           0.32820    0.05574   5.888 0.000000425 ***
## X3           3.87499    2.32962   1.663    0.103
## X5           3.87052    4.80772   0.805    0.425
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 190.1 on 46 degrees of freedom
## Multiple R-squared:  0.6074, Adjusted R-squared:  0.5818
## F-statistic: 23.72 on 3 and 46 DF,  p-value: 0.00000001993
```

iii. X1 en función de la variable predictora X2 y X3

En un tercer acercamiento a encontrar nuestro mejor MLR, seleccionaremos las variables X2 y X3 ya que presentan la mayor correlación con X1.

```
modelo_lineal_3 <- lm(X1 ~ X2 + X3, data = data)

# Resumen del modelo
options(scipen = 999)
summary(modelo_lineal_3)
```

```
##
## Call:
## lm(formula = X1 ~ X2 + X3, data = data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -331.08 -112.81  -35.76   79.31  673.81
##
## Coefficients:
##              Estimate Std. Error t value    Pr(>|t|)
## (Intercept) 350.8865    79.2530   4.427 0.000056535 ***
## X2           0.3355     0.0548   6.122 0.000000176 ***
## X3           4.2470     2.2748   1.867   0.0681 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 189.4 on 47 degrees of freedom
## Multiple R-squared:  0.6018, Adjusted R-squared:  0.5849
## F-statistic: 35.52 on 2 and 47 DF,  p-value: 0.0000000003996
```

Basándonos en el valor de R cuadrado ajustado, tomaremos como nuestro modelo aquel que utiliza X2 y X3 para encontrar X1, y en base a este haremos el análisis estadístico. Es decir, no es necesario tomar todas las variables como predictoras.

iv. Seleccionar el Modelo Lineal

El R cuadrado ajustado proporciona una medida de la calidad del modelo ajustado teniendo en cuenta el número de variables en el modelo. Un valor más alto indica un mejor ajuste, razón por la cual escogemos como mejor MLR nuestra opción (iii), ya que brinda un valor de R cuadrado ajustado igual a 0.5848857.

```
# Obtener el R cuadrado ajustado
r_cuadrado_ajustado_1 <- summary(modelo_lineal_1)$adj.r.squared
r_cuadrado_ajustado_2 <- summary(modelo_lineal_2)$adj.r.squared
r_cuadrado_ajustado_3 <- summary(modelo_lineal_3)$adj.r.squared

# Imprimir el resultado
cat("El R cuadrado ajustado para el modelo (i) es:", r_cuadrado_ajustado_1, "\n\n")
## El R cuadrado ajustado para el modelo (i) es: 0.5591831
cat("El R cuadrado ajustado para el modelo (ii) es:", r_cuadrado_ajustado_2, "\n\n")
## El R cuadrado ajustado para el modelo (ii) es: 0.5817544
cat("El R cuadrado ajustado para el modelo (iii) es:", r_cuadrado_ajustado_3, "\n\n")
## El R cuadrado ajustado para el modelo (iii) es: 0.5848857
```


iv. Evaluar el Modelo Lineal

Se establece un escenario de entrenamiento y validación para evaluar un modelo lineal. El conjunto de entrenamiento se crea seleccionando aleatoriamente el 60% de los índices entre 1 y 50, y el conjunto de validación se forma excluyendo las filas correspondientes al conjunto de entrenamiento.

Training

```
# Establecer una semilla para reproducibilidad
set.seed(1)

# Seleccionar aleatoriamente el 60% de los índices entre 1 y 50
train.index <- sample(1:50, 0.6 * 50)

# Crear el conjunto de entrenamiento (train.df) seleccionando las filas correspondientes a los índices seleccionados

train.df <- data[train.index, c("X1", "X2", "X3")]

# Crear el conjunto de validación
valid.df <- data[-train.index, c("X1", "X2", "X3")]
```

Testing

La función `predict` se utiliza para generar predicciones del modelo lineal (`modelo_lineal_3`) utilizando el conjunto de validación (`valid.df`). Las predicciones se almacenan en la variable `x1.lm.pred`.

Por su parte, los residuales representan la diferencia entre los valores observados y los valores predichos por el modelo. Se crea un data frame que contiene tres columnas: “Predicted” con las predicciones, “Actual” con los valores reales y “Residual” con los residuales; los cuales son el 40% de los datos de testing.

```
x1.lm.pred <- predict(modelo_lineal_3, valid.df)

options(scipen = 999)

some.residuals <- valid.df$X1[1:20]-x1.lm.pred[1:20]

data.frame("Predicted" = x1.lm.pred[1:20], "\nActual" = valid.df$X1[1:20], "\nResidual" = some.residuals)
```

	##	Predicted	X.Actual	X.Residual
##	2	558.2438	494	-64.24379
##	5	745.1308	773	27.86925
##	8	525.2539	546	20.74611
##	11	545.4892	506	-39.48917
##	13	574.3187	541	-33.31874
##	16	501.0273	371	-130.02735
##	17	513.6335	457	-56.63351
##	19	608.1212	570	-38.12118
##	24	771.5735	547	-224.57351
##	28	832.5282	867	34.47181
##	30	576.2380	462	-114.23805
##	32	878.0239	805	-73.02389
##	35	748.6614	919	170.33864

Examen Argumentativo

```
## 37 785.9120      657 -128.91197
## 38 745.1857     1419  673.81429
## 42 715.1352      815   99.86476
## 44 678.7632      936  257.23682
## 45 637.4416      863  225.55845
## 46 927.9879      783 -144.98790
## 50 1048.5073     940 -108.50729
```

Analizar Veracidad del Modelo

Ahora se medirán las métricas de precisión entre las predicciones del modelo. En general, valores más bajos en estas métricas indican un mejor rendimiento del modelo. Esto nos indica, especialmente basándonos en el RMSE, que tenemos un modelo que relativamente explica el comportamiento de X_1 ; no obstante requiere de mayor precisión.

```
accuracy(x1.lm.pred, valid.df$X1)

##                ME      RMSE      MAE      MPE      MAPE
## Test set 17.69119 194.8141 133.2988 -3.117867 17.44525
```

IV. Conclusiones

En el presente reporte se analizó un dataset, en el cual, basándonos en las correlaciones observadas con este dataset, podemos concluir que factores como la tasa de criminalidad violenta reportada (X2) y ciertos niveles educativos (X4 y X7) pueden influir en la tasa total general de criminalidad reportada por cada millón de residentes.

Por su parte, empleando el método de las estimaciones por mínimos cuadrados, y MLR se obtuvo que:

$$X1 = 350.8865 + 0.3355 x2 + 4.2470 x3 + \epsilon$$

Donde, el R cuadrado ajustado para el modelo (iii) es: 0.5848857 y el valor de RMSE es de: 194.8141

Se evaluaron varios modelos de regresión lineal múltiple, y el R cuadrado ajustado se utilizó como indicador de ajuste del modelo a los datos. El modelo (iii) presentó el R cuadrado ajustado más alto, indicando que explica una mayor proporción de la variabilidad en los datos en comparación con los modelos (i) y (ii).

El valor de RMSE proporciona una medida de la dispersión de las predicciones, con un valor de 194.8141 en este caso. Un R cuadrado ajustado más alto sugiere un mejor ajuste del modelo a los datos. En este caso, el modelo (iii) tiene el R cuadrado ajustado más alto, seguido por el modelo (ii) y luego el modelo (i).

Es por esta razón que seleccionamos las variables X2 y X3 ya que presentan la mayor correlación con X1, 0.75650513 y 0.5331978 respectivamente. Estas variables fueron elegidas debido a sus altas correlaciones con la variable de salida X1.