TECNOLÓGICO NACIONAL DE MEXICO
INSTITUTO TECNOLOGICO DE TIJUANA

SUBDIRECCIÓN ACADÉMICA

DEPARTAMENTO DE INGENIERÍA EN SISTEMAS COMPUTACIONALES

SEMESTRE FEBRERO-JUNIO 2022

MATERIA:

Minería de datos.

UNIDAD 3

Practica 2

Regresión lineal

DOCENTE:

JOSE CHRISTIAN ROMERO HERNANDEZ

ALUMNO:

López Higuera Saúl Alfredo #18210493

Munguía silva Edgar Geovanny #17212344

Tijuana BC 18 de mayo del 2022

Introducción. Primero, necesitamos cargar el CSV (lo proporcionó el profesor) y luego comenzaremos a analizar los datos, una vez que los datos estén completamente cargados, procedemos a convertir los estados a datos categóricos en números, luego dividimos el marco de datos en dos con una semilla aleatoria, de esta manera, los datos se distribuyen aleatoriamente. Hice algunos cambios menores en el código, por ejemplo, decidí usar file.choose en lugar del código provisto, porque es más fácil para mí usarlo de esta manera.

Código.

```
# Importing the dataset
dataset <- read.csv(file.choose())
# Encoding categorical data
dataset$State = factor(dataset$State,
                       levels = c('New York', 'California',
'Florida'),
                       labels = c(1,2,3))
dataset
# Splitting the dataset into the Training set and Test set
# Install.packages('caTools')
install.packages('caTools')
library(caTools)
set.seed(123)
split <- sample.split(dataset$Profit, SplitRatio = 0.8)
training_set <- subset(dataset, split == TRUE)
test_set <- subset(dataset, split == FALSE)

# Fitting Multiple Linear Regression to the Training set
#regressor = lm(formula = Profit ~ R.D.Spend + Administration +
Marketing.Spend + State)
regressor = lm(formula = Profit ~ .,
               data = training_set )
summary(regressor)
```

Salida. Estos son los resultados usando la regresión.

```
Call:
lm(formula = Profit ~ ., data = training_set)

Residuals:
```

```
   Min     1Q Median     3Q    Max
-33128  -4865      5   6098  18065

Coefficients:
                  Estimate Std. Error t value Pr(>|t|)
(Intercept)      4.965e+04  7.637e+03   6.501 1.94e-07 ***
R.D.Spend        7.986e-01  5.604e-02  14.251 6.70e-16 ***
Administration  -2.942e-02  5.828e-02  -0.505    0.617
Marketing.Spend  3.268e-02  2.127e-02   1.537    0.134
State2           1.213e+02  3.751e+03   0.032    0.974
State3           2.376e+02  4.127e+03   0.058    0.954
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 9908 on 34 degrees of freedom
Multiple R-squared:  0.9499,     Adjusted R-squared:  0.9425
F-statistic:   129 on 5 and 34 DF,  p-value: < 2.2e-16
```

Predicciones. En esta sección, mostrará las predicciones que tendría cada campo en el marco de datos

```
# Prediction the Test set results
y_pred = predict(regressor, newdata = test_set)
y_pred
```

Salida.

```
> y_pred = predict(regressor, newdata = test_set)
> y_pred
       4         5         8        11        16        20
21        24        31        32
173981.09 172655.64 160250.02 135513.90 146059.36 114151.03
117081.62 110671.31  98975.29  96867.03
```

Preparar los datos para usar backwards elimination.
Antes de usar la eliminación hacia atrás o backwards elimination, necesitamos optimizar el dataframe, lo que vamos a hacer es reducir los campos a campos clave, solo para que sea más fácil para nosotros trabajar con los datos.

```
# Assigment: visualize the siple liner regression model with
```

```
R.D.Spend

# Building the optimal model using Backward Elimination
regressor = lm(formula = Profit ~ R.D.Spend + Administration +
Marketing.Spend + State,
               data = dataset )
summary(regressor)
```

Salida.

```
Call:
lm(formula = Profit ~ R.D.Spend + Administration + Marketing.Spend
+
    State, data = dataset)

Residuals:
   Min     1Q Median     3Q    Max
-33504  -4736     90   6672  17338

Coefficients:
                  Estimate Std. Error t value Pr(>|t|)
(Intercept)      5.008e+04  6.953e+03   7.204 5.76e-09 ***
R.D.Spend        8.060e-01  4.641e-02  17.369  < 2e-16 ***
Administration  -2.700e-02  5.223e-02  -0.517    0.608
Marketing.Spend  2.698e-02  1.714e-02   1.574    0.123
State2           4.189e+01  3.256e+03   0.013    0.990
State3           2.407e+02  3.339e+03   0.072    0.943
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 9439 on 44 degrees of freedom
Multiple R-squared:  0.9508,     Adjusted R-squared:  0.9452
F-statistic: 169.9 on 5 and 44 DF,  p-value: < 2.2e-16
```

Una vez realizada la reducción, se procede a utilizar la función de eliminación hacia atrás.

```
# Homework analise the follow atomation backwardElimination
function
backwardElimination <- function(x, sl) {
  numVars = length(x)
```

```
  for (i in c(1:numVars)){
    regressor = lm(formula = Profit ~ ., data = x)
    maxVar = max(coef(summary(regressor))[c(2:numVars),
"Pr(>|t|)"])
    if (maxVar > sl){
      j = which(coef(summary(regressor))[c(2:numVars), "Pr(>|t|)"]
== maxVar)
      x = x[, -j]
    }
    numVars = numVars - 1
  }
  return(summary(regressor))
}

SL = 0.05
#dataset = dataset[, c(1,2,3,4,5)]
training_set
```

Salida.

```
R.D.Spend Administration Marketing.Spend State    Profit
1  165349.20      136897.80       471784.10     1 192261.83
2  162597.70      151377.59       443898.53     2 191792.06
3  153441.51      101145.55       407934.54     3 191050.39
6  131876.90       99814.71       362861.36     1 156991.12
7  134615.46      147198.87       127716.82     2 156122.51
9  120542.52      148718.95       311613.29     1 152211.77
10 123334.88      108679.17       304981.62     2 149759.96
12 100671.96       91790.61       249744.55     2 144259.40
13  93863.75      127320.38       249839.44     3 141585.52
14  91992.39      135495.07       252664.93     2 134307.35
15 119943.24      156547.42       256512.92     3 132602.65
17  78013.11      121597.55       264346.06     2 126992.93
18  94657.16      145077.58       282574.31     1 125370.37
19  91749.16      114175.79       294919.57     3 124266.90
22  78389.47      153773.43       299737.29     1 111313.02
23  73994.56      122782.75       303319.26     3 110352.25
25  77044.01       99281.34       140574.81     1 108552.04
26  64664.71      139553.16       137962.62     2 107404.34
27  75328.87      144135.98       134050.07     3 105733.54
28  72107.60      127864.55       353183.81     1 105008.31
```

```
29  66051.52       182645.56       118148.20     3 103282.38
30  65605.48       153032.06       107138.38     1 101004.64
33  63408.86       129219.61        46085.25     2  97427.84
34  55493.95       103057.49       214634.81     3  96778.92
35  46426.07       157693.92       210797.67     2  96712.80
36  46014.02        85047.44       205517.64     1  96479.51
37  28663.76       127056.21       201126.82     3  90708.19
38  44069.95        51283.14       197029.42     2  89949.14
39  20229.59        65947.93       185265.10     1  81229.06
40  38558.51        82982.09       174999.30     2  81005.76
41  28754.33       118546.05       172795.67     2  78239.91
42  27892.92        84710.77       164470.71     3  77798.83
43  23640.93        96189.63       148001.11     2  71498.49
44  15505.73       127382.30        35534.17     1  69758.98
45  22177.74       154806.14        28334.72     2  65200.33
46   1000.23       124153.04         1903.93     1  64926.08
47   1315.46       115816.21       297114.46     3  49490.75
48      0.00       135426.92            0.00     2  42559.73
49    542.05        51743.15            0.00     1  35673.41
50      0.00       116983.80        45173.06     2  14681.40
>
```

Resultados.

Y por último, pero no menos importante, usaremos el siguiente código para mostrar los resultados, mostrará muchos datos útiles, como la moda, la mediana y el promedio.

```
backwardElimination(training_set, SL)
```

Salida.

```
> backwardElimination(training_set, SL)

Call:
lm(formula = Profit ~ ., data = x)

Residuals:
   Min     1Q Median     3Q    Max
-34334  -4894   -340   6752  17147

Coefficients:
```

```
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 4.902e+04  2.748e+03   17.84   <2e-16 ***
R.D.Spend   8.563e-01  3.357e-02   25.51   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 9836 on 38 degrees of freedom
Multiple R-squared:  0.9448,     Adjusted R-squared:  0.9434
F-statistic: 650.8 on 1 and 38 DF,  p-value: < 2.2e-16
```