TECNOLÓGICO NACIONAL DE MÉXICO

INSTITUTO TECNOLÓGICO DE TIJUANA

SUBDIRECCIÓN ACADÉMICA

DEPARTAMENTO DE SISTEMAS Y COMPUTACIÓN

NOMBRE DE LOS ALUMNOS:

RAYMUNDO HIRALES LAZARENO (N. CONTROL: 17212339)

GALAVIZ LONA OSCAR EDUARDO (N.CONTROL: 17212993)

Carrera: Ingeniería Informática

Semestre: 9no

MATERIA: Minería de datos

PROFESOR: JOSE CHRISTIAN ROMERO HERNANDEZ

TRABAJOS: Practica 2

FECHA: 26/11/21

Desarrollo

Aqui comenzamos con la carga del archivo csv que se utilizara para el analisis de datos de esta practica, una vez cargados procedemos a que los estados sean convertidos a datos categoricos en este caso numeros, despues dividimos el dataframe en dos con un semilla de aleatoriedad para que los datos se repartan de manera aleatoria

```
getwd()
setwd("/home/chris/Documents/itt/Enero_Junio_2020/Mineria_de_datos/DataMining/Mach
ineLearning/MultipleLinearRegression")
getwd()

# Importing the dataset
dataset <- read.csv('50_Startups.csv')

# 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')
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)
```

Por ultimo los resultado del modelo que mediante la regresion

```
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

Aqui nos muestran las predicciones que tendria cada uno de los campos del dataframe

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

```
> 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
```

Optimizacion del modelo para usar backward elimanation

En este apartado empezamos a optimizar el dataframe para la utilizacion del backward elimanation reduciendo el dataframe a solo unos cuantos campos claves del dataframe que se utilizaran para este analisis

```
# 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)
```

```
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
```

```
regressor = lm(formula = Profit ~ R.D.Spend + Administration + Marketing.Spend,
               data = dataset )
summary(regressor)
```

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

Residuals:
   Min     1Q Median     3Q    Max
-33534  -4795     63   6606  17275

Coefficients:
                 Estimate Std. Error t value Pr(>|t|)
(Intercept)      5.012e+04  6.572e+03   7.626 1.06e-09 ***
R.D.Spend        8.057e-01  4.515e-02  17.846  < 2e-16 ***
Administration  -2.682e-02  5.103e-02  -0.526    0.602
Marketing.Spend  2.723e-02  1.645e-02   1.655    0.105
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 9232 on 46 degrees of freedom
Multiple R-squared:  0.9507,    Adjusted R-squared:  0.9475
F-statistic:   296 on 3 and 46 DF,  p-value: < 2.2e-16
```

```
regressor = lm(formula = Profit ~ R.D.Spend + Marketing.Spend,
               data = dataset )
summary(regressor)

regressor = lm(formula = Profit ~ R.D.Spend + Marketing.Spend,
               data = dataset )
summary(regressor)
```

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

Residuals:
   Min     1Q Median     3Q    Max
-33645  -4632   -414   6484  17097

Coefficients:
                 Estimate Std. Error t value Pr(>|t|)
(Intercept)     4.698e+04  2.690e+03  17.464   <2e-16 ***
R.D.Spend       7.966e-01  4.135e-02  19.266   <2e-16 ***
Marketing.Spend 2.991e-02  1.552e-02   1.927     0.06 .
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 9161 on 47 degrees of freedom
Multiple R-squared:  0.9505,    Adjusted R-squared:  0.9483
F-statistic: 450.8 on 2 and 47 DF,  p-value: < 2.2e-16
```

```
y_pred = predict(regressor, newdata = test_set)
y_pred
```

```
> y_pred
       4        5        8       11       16       20       21       24       31       32
173441.31 171127.62 160455.74 135011.91 146032.72 115816.42 116650.89 109886.19  99085.22  98314.55
```

uso de backwardelimination

una vez reducido el dataframe procedemos a realizar el uso de la funcion de backwardelimination. creamos la funcion para que se realice en el dataframe que reducimos

```
# 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
```

```
> training_set
   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
```

resultados del backwardelimination

Aqui se nos muestran los resultados como el analisis anterior pero con la funcion del analisis de backwardeliminatio y nos mostrara una serie de resultados estadisticos como la media, la mediana, el cuartil, el error estandar entre otros que me muestra

```
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
```