Tecnológico Nacional de México
Instituto Tecnológico de Tijuana

Subdirección Académica
Departamento de Sistemas y Computación
Ingeniería en Sistemas Computacionales
Semestre: AGOSTO-DICIEMBRE 2021

MINERÍA DE DATOS

BDD-1703SC9A

Práctica 1

Landa Alvarez Ariel Nicolas 17211531
Ceron Uribe Arturo #17211506

MC. JOSE CHRISTIAN ROMERO HERNANDEZ

Campus Tomas Aquino

Analizar el código correspondiente a la visualización de datos de modelo de machine learning regresión lineal este código esta en mi repositorio aquí dejo el enlace.
https://github.com/jcromerohdz/DataMining/tree/master/MachineLearning/SimpleLinearRegression

El primer paso a realizar es la lectura de los datos, para esto se es necesario establecer el directorio donde se encuentra el script, junto con el csv, para ello utilizaremos la función getwd() y con ello simplemente importamos el csv sin necesidad de escribir su ubicación en el sistema.

```
getwd()
# Importing the dataset
dataset <- read.csv('Salary_Data.csv')
```

Para separar el dataset en entrenamiento y prueba, utilizamos la función split que se encargará de separar el dataset en dos partes con un ratio de 66% y 33% aproximadamente y utilizando subset podremos asignar cada parte a un dataset correspondiente.

```
# Splitting the dataset into the Training set and Test set
split <- sample.split(dataset$Salary, SplitRatio = 2/3)
training_set <- subset(dataset, split == TRUE)
test_set <- subset(dataset, split == FALSE)
```

Para hacer uso de un modelo lineal simple como es regresión lineal, utilizamos la función lm especificando las variables relacionadas en este caso salario y años de experiencia y el set de datos correspondiente.

```
# Fitting Simple Linear Regression to the Training set
regressor = lm(formula = Salary ~ YearsExperience,
               data = dataset)
summary(regressor)
```

summary retorna el resultado siguiente. Estos son todos los valores generados por R que son necesarios para el modelo.

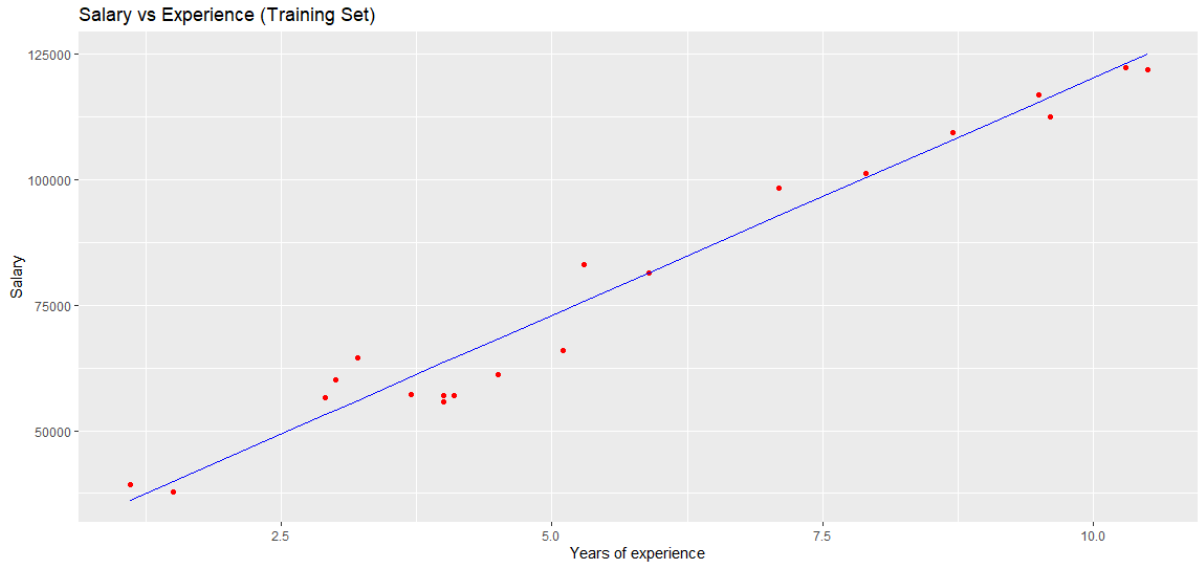
```
Call:
lm(formula = Salary ~ YearsExperience, data = dataset)

Residuals:
    Min      1Q  Median      3Q     Max
-7958.0 -4088.5  -459.9  3372.6 11448.0

Coefficients:
               Estimate Std. Error t value Pr(>|t|)
(Intercept)      25792.2     2273.1   11.35 5.51e-12 ***
YearsExperience   9450.0      378.8   24.95  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 5788 on 28 degrees of freedom
Multiple R-squared:  0.957,      Adjusted R-squared:  0.9554
F-statistic: 622.5 on 1 and 28 DF,  p-value: < 2.2e-16
```

Teniendo ya nuestro modelo de predicción listo, podemos proceder a darle un entrenamiento utilizando datos, utilizando el formato siguiente, podemos asignar datos que respeten el modelo anteriormente.

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

El valor que generamos lo podemos asignar en un ggplot() como la variable Y ya que esta será la predicción que genera el modelo. En el caso a continuación se utiliza el dataset training_set para mostrar la línea que se genera, de este modo se puede hacer la comparación entre la línea de predicción y los puntos reales

```
# Visualising the Training set results
library(ggplot2)
ggplot() +
  geom_point(aes(x=training_set$YearsExperience, y=training_set$Salary),
             color = 'red') +
  geom_line(aes(x = training_set$YearsExperience, y = predict(regressor, newdata
= training_set)),
            color = 'blue') +
  ggtitle('Salary vs Experience (Training Set)') +
  xlab('Years of experience') +
  ylab('Salary')
```



Este caso es similar al anterior, con la diferencia de que los datos reales comparados son del dataset test_set, el cual podemos comparar con la línea de regresión generada del dataset training_set.
Este es el resultado final que se desea obtener.

```
# Visualising the Test set results
ggplot() +
  geom_point(aes(x=test_set$YearsExperience, y=test_set$Salary),
             color = 'red') +
  geom_line(aes(x = training_set$YearsExperience, y = predict(regressor, newdata
= training_set)),
            color = 'blue') +
  ggtitle('Salary vs Experience (Test Set)') +
  xlab('Years of experience') +
  ylab('Salary')
```