



TECNOLÓGICO  
NACIONAL DE MÉXICO



**SEP**  
SECRETARÍA DE  
EDUCACIÓN PÚBLICA

**National Technology of Mexico  
Technological Institute of Tijuana**

ACADEMIC SUBDIRECTION  
Systems and Computing Department

SEMESTER  
February – June 2021

CAREER  
Information and Communication Technologies Engineer

SUBJECT AND KEY:  
Data Mining BDD-1703TI9A

STUDENT'S NAME AND REGISTRATION:  
Camacho Manabe Juan Daniel 17210534  
Velázquez Farrera César Alejandro 17212937

NAME OF THE JOB:  
Practice #1

UNIT TO BE EVALUATED  
Unit III

TEACHER'S NAME:  
Mc. José Christian Romero Hernández

## Practice #1

### Instructions:

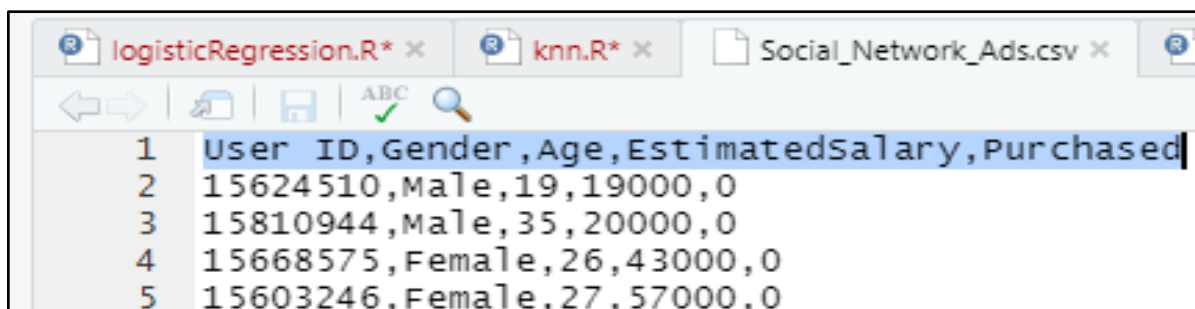
Make the analysis corresponding to the logistic regression R script which must be documented in its repository, putting in it its visual results and its detailed description of its observations as well as the source of the code.

We can define the path we want to work on.

```
getwd()
setwd("/home/chris/Documents/itt/Enero_Junio_2020/Mineria_de_datos/DataMining/MachineLearning/LogisticRegression")
getwd()
```

We look for our data file on the computer with the help of the file explorer, if we are working in the folder where the file is located, we simply load it with the name of the file.

```
dataset = read.csv(file.choose())
dataset = read.csv('Social_Network_Ads.csv')
```



	User ID,Gender,Age,EstimatedSalary,Purchased
2	15624510,Male,19,19000,0
3	15810944,Male,35,20000,0
4	15668575,Female,26,43000,0
5	15603246,Female,27,57000,0

Since the data file has columns that will not be useful, we only select the columns that we want to work with.

```
dataset = dataset[3:5]
```

\$ Age	: int	19	35	26	27	19	27	27	32	25	35	...
\$ EstimatedSalary:	int	19000	20000	43000	57000	76000	58000	84000	150000	330...		
\$ Purchased	: int	0	0	0	0	0	0	1	0	0	...	

We load the library "caTools" that has several functions for statistics.

With the function "set.seed (n)" we sow a randomness seed where "n" is the starting point used in the generation of the sequence of random numbers.

```
library(caTools)
set.seed(123)
```

Next, we divide the data in the "Purchased" column, with a ratio of 0.75, that is, 75% of the data will be taken as true and the remaining 25% as false.

Later we take the values of the division of the data that were true and we use them for training, the rest of the data was labeled as false will be used to test the effectiveness of the trained model with the data labeled as true.

```
split = sample.split(dataset$Purchased, SplitRatio = 0.75)
training_set = subset(dataset, split == TRUE)
test_set = subset(dataset, split == FALSE)
```

For training and testing it is necessary to normalize the data, so we use the "scale" function. The characteristic scale is a method used to normalize the range of independent variables or characteristics of the data. In data processing, it is also known as data normalization and is usually done during the data pre-processing step.

```
training_set[, 1:2] <- scale(training_set[, 1:2])
test_set[, 1:2] <- scale(test_set[, 1:2])
```

We adopt the logistic regression to the training data set, that is, the classification with the "glm" method which is used to fit linear models.

```
classifier = glm(formula = Purchased ~ .,
                 family = binomial,
                 data = training_set)
```

Obtaining the probability of predicting correct classifications with the test data. We print the prediction. We evaluate if the probability of the prediction is between 0.5, 1 and 0 and print the probability.

```
prob_pred = predict(classifier, type = 'response', newdata = test_set[-3])
prob_pred
y_pred = ifelse(prob_pred > 0.5, 1, 0)
y_pred
```

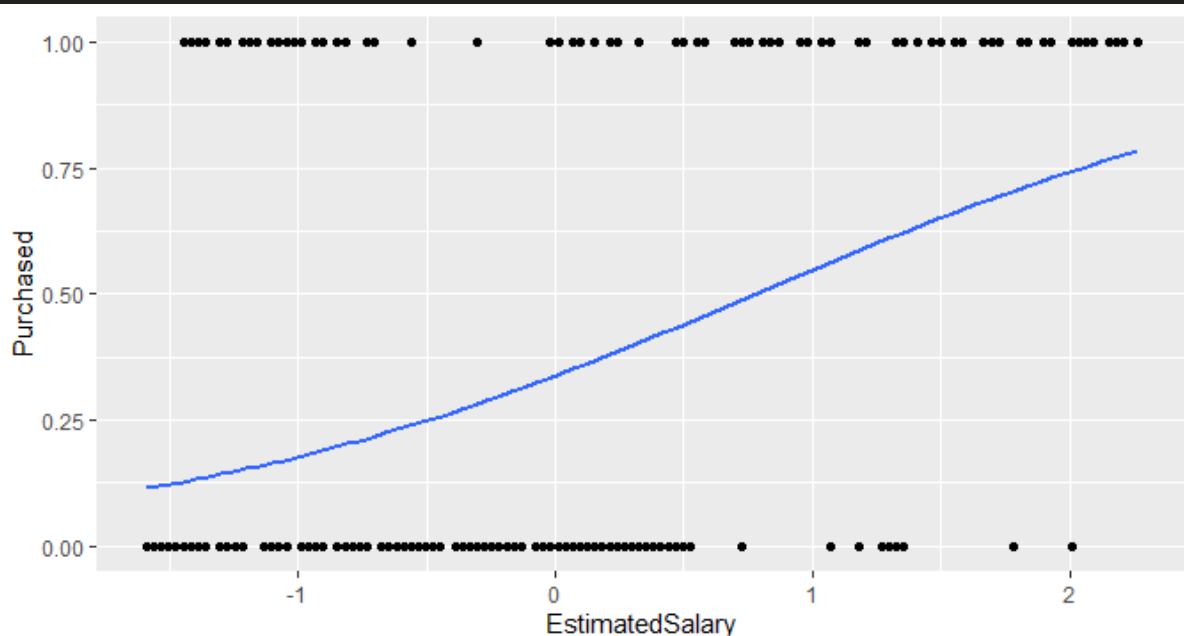
Guardamos los datos resultantes en una tabla con los datos de prueba y las predicciones correspondientes, es decir, creamos una matriz de confusión. Luego la imprimimos.

```
cm = table(test_set[, 3], y_pred)
cm
```

We load the graphics library "ggplot2".

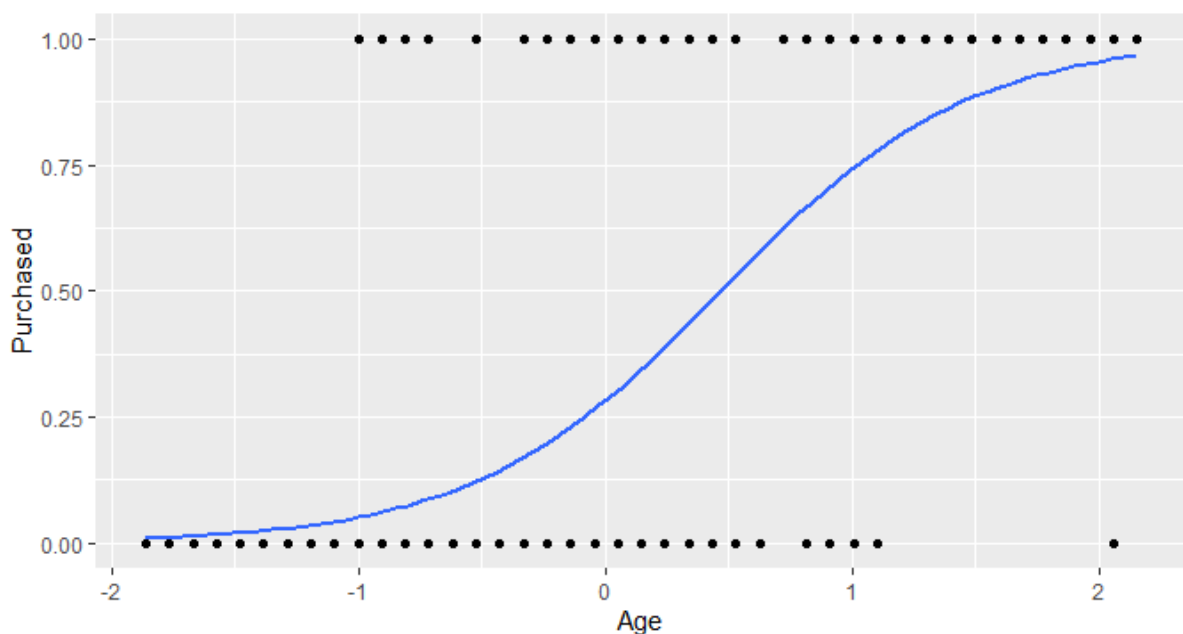
We graph with the training data using the estimated salary and purchase columns, we add a point geometry and a smooth geometry that together with the "glm" method which is used to fit generalized or specified linear models giving a symbolic description of the linear predictor and a description of the error distribution. Also a method that lists additional arguments in this case which is a regression of the binomial family.

```
library(ggplot2)
ggplot(training_set, aes(x=EstimatedSalary, y=Purchased)) + geom_point() +
  stat_smooth(method="glm", method.args=list(family="binomial"), se=FALSE)
```



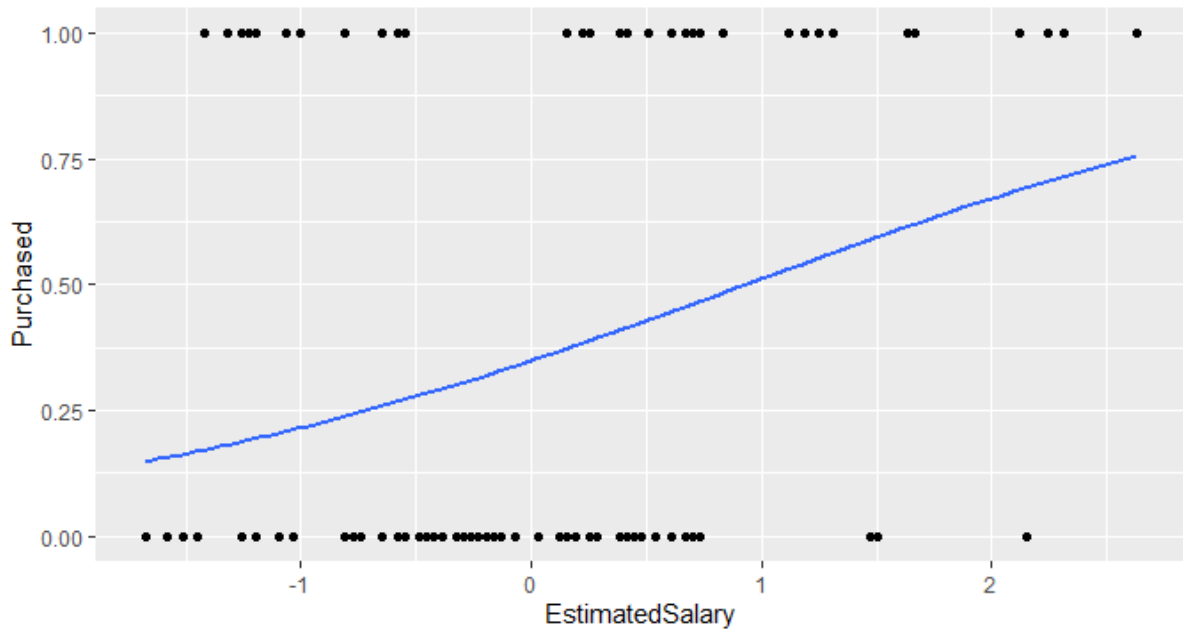
We graph with the training data using the age and purchase columns, we add a point geometry and a smooth geometry that together with the "glm" method which is used to fit generalized or specified linear models giving a symbolic description of the linear predictor and a description of the error distribution. Also a method that lists additional arguments in this case which is a regression of the binomial family.

```
ggplot(training_set, aes(x=Age, y=Purchased)) + geom_point() +  
  stat_smooth(method="glm", method.args=list(family="binomial"), se=FALSE)
```



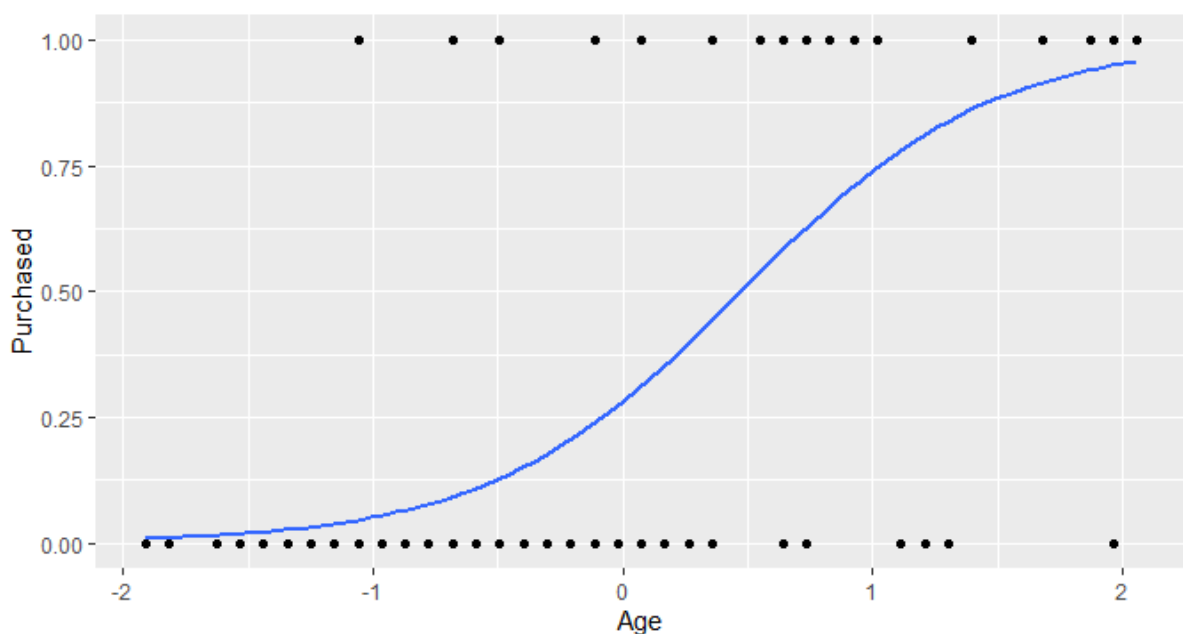
We graph with the test data using the estimated salary and purchase columns, we add a point geometry and a smooth geometry that together with the "glm" method which is used to fit generalized or specified linear models giving a symbolic description of the linear predictor and a description of the error distribution. Also a method that lists additional arguments in this case which is a regression of the binomial family.

```
ggplot(test_set, aes(x=EstimatedSalary, y=Purchased)) + geom_point() +  
  stat_smooth(method="glm", method.args=list(family="binomial"), se=FALSE)
```



We graph with the test data using the age and purchase columns, we add a point geometry and a smooth geometry that together with the "glm" method which is used to fit generalized or specified linear models giving a symbolic description of the linear predictor and a description of the error distribution. Also a method that lists additional arguments in this case which is a regression of the binomial family.

```
ggplot(test_set, aes(x=Age, y=Purchased)) + geom_point() +
  stat_smooth(method="glm", method.args=list(family="binomial"), se=FALSE)
```



To continue we will need a special package that contains the "ElemStatLearn" library, since the package is obsolete for new versions of R and does not come by default in the RStudio tools, it is necessary to download and install it manually. The file will be found in the following link: <https://cran.r-project.org/src/contrib/Archive/ElemStatLearn/>, the installation will be done with the following command, substituting the appropriate file address.

```
install.packages("~/home/daniel-  
camacho/Descargas/ElemStatLearn_2015.6.26.tar.gz", repos=NULL,  
type="source")
```

1. Once the package is installed, we load the "ElemStatLearn" library.
2. With the command "set" we take the training data, then we create two variables "x1", "x2" where we use the function "seq" (sequence), we will define a minimum and maximum range, we will add a positive integer and multiply it by the length of "0.01"
3. Next we define a grid that we expand using the dimensions defined in "x1" and "x2".
4. We define the columns of the grid using a vector, which will be the year and estimated salary.
5. Next we test the logistic regression model but with the grid that contains the columns that we define.
6. We graph the data with a maximum of -1, we put a title to the graph that is inside (grid), we name the axes, we assign the limits for the axes keeping the data within the dimensions of the grid.
7. We place an outline and add a range of numbers conditional on the minimum and maximum numerical range of the grid.
8. We add points to the graph comparing the data of the year and estimated salaries with the predictions of the logistic regression model, where if they are different, a red color will be assigned and if they are true a green one, in the case of errors, red points will be placed within the green area that has been defined as true sets and, on the contrary, a green point in the area defined as false.

```

library(ElemStatLearn)
set = training_set
X1 = seq(min(set[, 1]) - 1, max(set[, 1]) + 1, by = 0.01)
X2 = seq(min(set[, 2]) - 1, max(set[, 2]) + 1, by = 0.01)
grid_set = expand.grid(X1, X2)
colnames(grid_set) = c('Age', 'EstimatedSalary')
prob_set = predict(classifier, type = 'response', newdata = grid_set)
y_grid = ifelse(prob_set > 0.5, 1, 0)
plot(set[, -3],
      main = 'Logistic Regression (Training set)',
      xlab = 'Age', ylab = 'Estimated Salary',
      xlim = range(X1), ylim = range(X2))
contour(X1, X2, matrix(as.numeric(y_grid), length(X1), length(X2)), add =
TRUE)
points(grid_set, pch = '.', col = ifelse(y_grid == 1, 'springgreen3',
'tomato'))
points(set, pch = 21, bg = ifelse(set[, 3] == 1, 'green4', 'red3'))

```



In the same way as in the previous code, the evaluation and classification of the data is done, only this time, for the graph it will not be the training data but the test data, for which in the "set" command the data is assigned of the variable "test\_set" obtained in the division of the data corresponding to 25% of the totality of the dataset that contains the three initial columns.

```

library(ElemStatLearn)
set = test_set
X1 = seq(min(set[, 1]) - 1, max(set[, 1]) + 1, by = 0.01)
X2 = seq(min(set[, 2]) - 1, max(set[, 2]) + 1, by = 0.01)
grid_set = expand.grid(X1, X2)

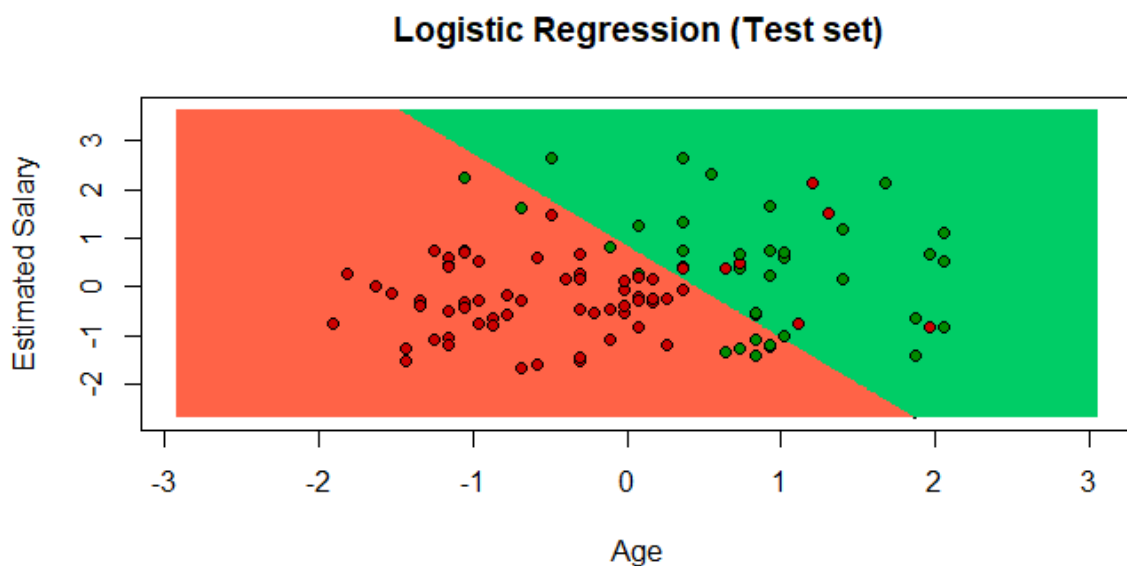
```



```

colnames(grid_set) = c('Age', 'EstimatedSalary')
prob_set = predict(classifier, type = 'response', newdata = grid_set)
y_grid = ifelse(prob_set > 0.5, 1, 0)
plot(set[, -3],
      main = 'Logistic Regression (Test set)',
      xlab = 'Age', ylab = 'Estimated Salary',
      xlim = range(X1), ylim = range(X2))
contour(X1, X2, matrix(as.numeric(y_grid), length(X1), length(X2)), add =
TRUE)
points(grid_set, pch = '.', col = ifelse(y_grid == 1, 'springgreen3',
'tomato'))
points(set, pch = 21, bg = ifelse(set[, 3] == 1, 'green4', 'red3'))

```



## Conclusions and observations

En base a las observaciones, el primer método es el más sencillo de implementar en términos de sintaxis y comprensión del lenguaje R. Sin embargo, al momento de representar los resultados de las predicción (utilizando la matriz de confusión y ), no es fácil de entender a primera vista para los usuarios que desconocen el tema.

Con los resultados obtenidos con las predicciones, se puede concluir que las columnas de edad ("age") tiene mayor correlación que las columnas de salario estimado ("EstimatedSalary") y Compras Realizadas ("Purchased"). Esto es evidente en las gráficas de puntos con el método "gml" por la línea azul, que representa el predictor lineal.

The second method, with the use of the "ElemStatLearn" library, can generate a graph much easier to understand, at the cost of complex syntax. It can be seen in the second graph of the second method, the red points represent the people who did not buy the product and the green ones those who did.

We can conclude that the older the potential customer is, the more likely they are to sell the product. Unlike the customer being younger, there is less chance that he will purchase the product.

# Bibliography

1. jcromerohdz. (2020). LogisticRegression. 29/05/30, de GitHub Sitio web: <https://github.com/jcromerohdz/DataMining/tree/master/MachineLearning/LogisticRegression>
2. [https://cran-archive.r-project.org/web/checks/2020/2020-01-28\\_check\\_results\\_ElemStatLearn.html](https://cran-archive.r-project.org/web/checks/2020/2020-01-28_check_results_ElemStatLearn.html)
3. <https://cran.r-project.org/src/contrib/Archive/ElemStatLearn/>
4. <https://riptutorial.com/r/example/5556/install-package-from-local-source>
5. Reche, J. L. C. (2013). Regresión logística. Tratamiento computacional con R.
6. Aldas Manzano, J., & Uriel Jimenez, E. (2017). *Análisis multivariante aplicado con R*. Ediciones Paraninfo, SA.
7. Escobar Moreno, N. (2013). Análisis de Regresión Logística para Investigación de Mercados (Logistic Regression Analysis for Marketing Research). *Available at SSRN 2591101*.
8. Serna Pineda, S. C. (2009). Comparación de árboles de regresión y clasificación y regresión logística. *Facultad de Ciencias*.
9. Martínez-Toro, G. M., Rico-Bautista, D., & Romero-Riaño, E. (2019). Análisis comparativo de predicción dentro de bases de datos de cáncer: una aplicación de aprendizaje automático. *Revista Ibérica de Sistemas e Tecnologías de Informação*, (E17), 113-122.