



**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:  
Statistical Distributions

UNIT TO BE EVALUATED  
Unit II

TEACHER'S NAME:  
Mc. José Christian Romero Hernández

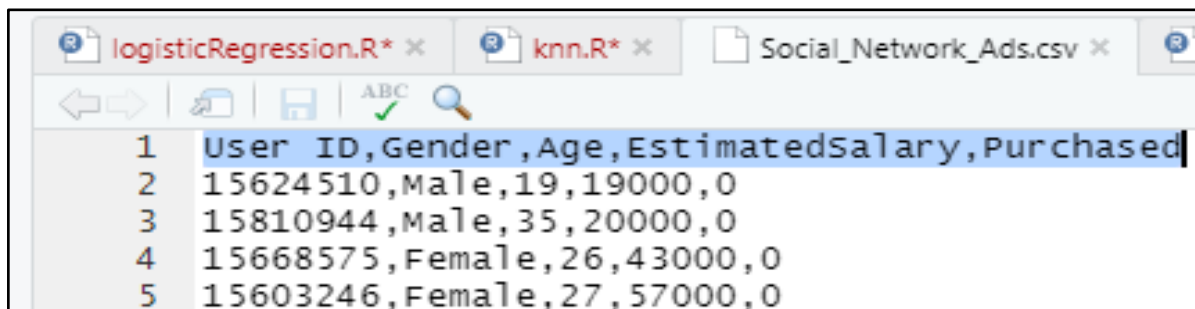
## Practice #2

### Instructions:

Make the analysis corresponding to the R script of K-Nearest Neighbors (K-NN) which must be documented in its repository by putting in it its visual results and its detailed description of its observations as well as the source of the code.

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
1	15624510	Male	19	19000	0
2	15810944	Male	35	20000	0
3	15668575	Female	26	43000	0
4	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	...	

From the three columns that we will use, we select the “Purchased” column from the dataset and with the “factor” function we categorize the values into two levels, in this case binary 0 and 1.

```
dataset$Purchased = factor(dataset$Purchased, levels = c(0, 1))
```

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[-3] = scale(training_set[-3])
test_set[-3] = scale(test_set[-3])
```

We loaded the “class” library, which includes several functions for classification, including nearest k-neighbor, learning vector quantization, and self-organizing maps.

We adapt K-NN to the training data set and prediction of the test set results

We create the model using the "knn" function included in the library, we pass it the training data, the test data, the variable where the prediction will be saved, the value of k and which data we want to evaluate, in this case the data where the probability is true.

```
library(class)
y_pred = knn(train = training_set[, -3],
             test = test_set[, -3],
             cl = training_set[, 3],
             k = 5,
             prob = TRUE)
```

We save the resulting data in a table with the test data and the corresponding predictions, that is, we create a confusion matrix.

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

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")
```

Una vez instalado el paquete, cargamos la librería "*ElemStatLearn*".

Con el comando "set" tomamos los datos de entrenamiento, después creamos dos variables "x1", "x2" en donde empleando la función "seq" (secuencia) definiremos un rango mínimo y máximo, le sumaremos un entero positivo y lo multiplicaremos por la longitud de "0.01"

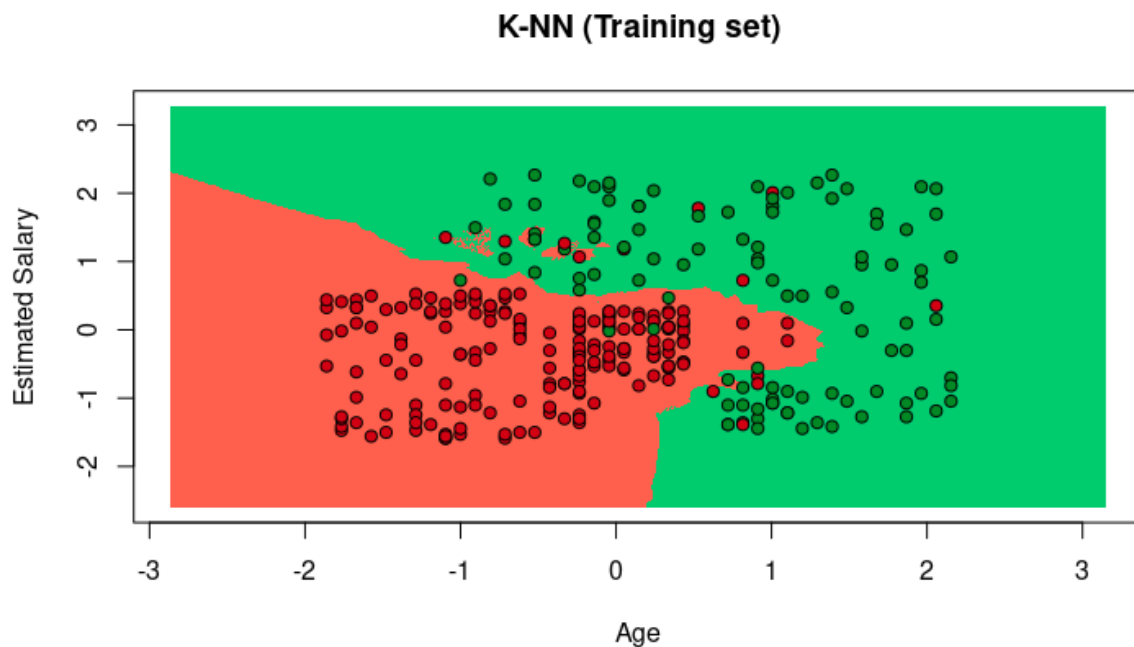
Next we define a grid that we expand using the dimensions defined in "x1" and "x2". We define the columns of the grid using a vector, which will be age and estimated salary.

Next, we test the knn model but with the grid that contains the columns that we define. We graph the data with a maximum of -3, we title 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.

We place an outline and add a range of numbers conditional on the minimum and maximum numerical range of the grid.

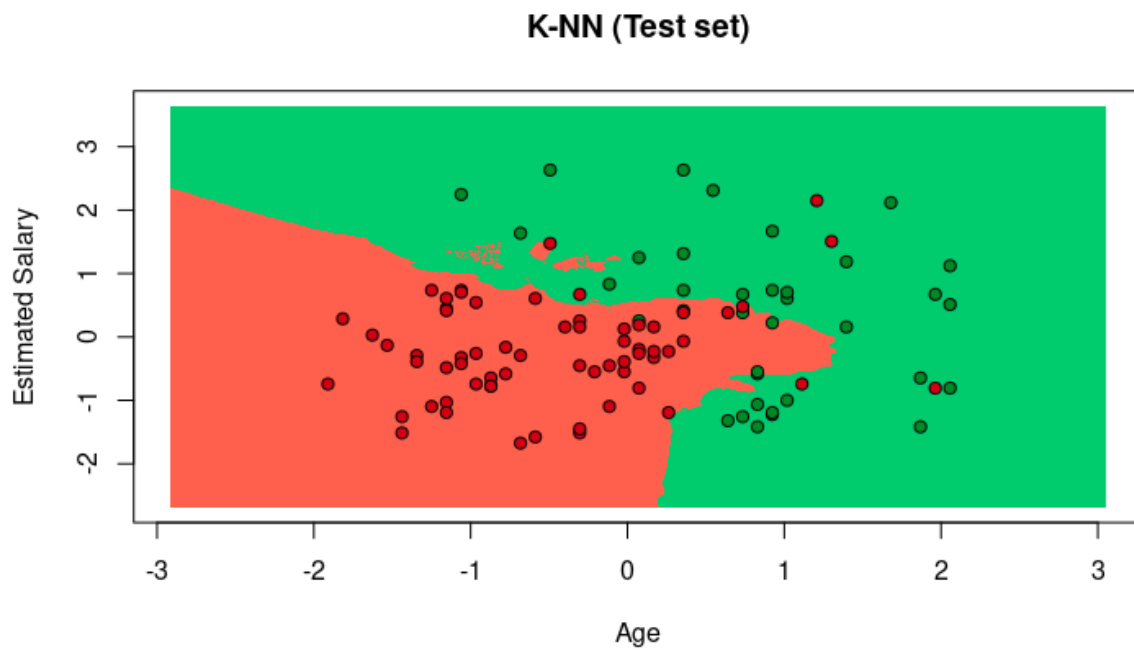
We add points to the graph comparing the data of age and estimated salary with the predictions of the KNN model, where if they are different, a red color will be assigned and if they are true a green one, in the case of errors, red points will be placed within the green area that has been defined as 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')
y_grid = knn(train = training_set[, -3], test = grid_set, cl =
training_set[, 3], k = 5)
plot(set[, -3],
      main = 'K-NN (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')
y_grid = knn(train = training_set[, -3], test = grid_set, cl =
training_set[, 3], k = 5)
plot(set[, -3],
      main = 'K-NN (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'))
```



## Conclusiones y observaciones

In the graphs, we can see that the KNN model is correct in most of the data. We can determine that the younger the buyer and the higher the earnings, he will buy the product. The purchase of the product is identified by the green color shown in the graph and the red color of those who did not purchase the product.

# Bibliography

1. jcromerohdz. (2020). KNN. 29/05/30, de GitHub Sitio web:  
<https://github.com/jcromerohdz/DataMining/tree/master/MachineLearning/KNN>
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. Guo, G., Wang, H., Bell, D., Bi, Y., & Greer, K. (2003, November). KNN model-based approach in classification. In *OTM Confederated International Conferences" On the Move to Meaningful Internet Systems"* (pp. 986-996). Springer, Berlin, Heidelberg.
6. Zhang, S., Li, X., Zong, M., Zhu, X., & Cheng, D. (2017). Learning k for knn classification. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 8(3), 1-19.
7. Xing, W., & Bei, Y. (2019). Medical health big data classification based on KNN classification algorithm. *IEEE Access*, 8, 28808-28819.
8. Rodríguez Vázquez, S., Martínez Borges, A. V., & Lorenzo Ginori, J. V. (2016). Clasificación de células cervicales mediante el algoritmo KNN usando rasgos del núcleo. *Revista Cubana de Ciencias Informáticas*, 10(1), 95-109.
9. De la Rosa, L. A. G. (2019). Sistema Clasificador por Knn de Vida Útil para Mermeladas de Fruta.
10. Ruiz Shulcloper, J., & Nolasco, J. A. (2011). Número Especial en Reconocimiento de Patrones, Minería de Datos y Aplicaciones.
11. Maíllo, J., Luengo, J., García, S., Herrera, F., & Triguero, I. (2018). Un enfoque aproximado para acelerar el algoritmo de clasificación Fuzzy kNN para Big Data. In *XVIII Conferencia de la Asociación Española para la Inteligencia Artificial (CAEPIA 2018) 23-26 de octubre de 2018 Granada, España* (pp. 1143-1148). Asociación Española para la Inteligencia Artificial (AEPIA).
12. Lozano Colomer, C., & Martínez de Ibarreta Zorita, C. (2019). Machine Learning I: Regresión y clasificación/Machine Learning I: Regression and classification.