Tecnológico Nacional de México
Instituto Tecnológico de Tijuana

Subdirección Académica
Departamento de Sistemas y Computación

Semestre:
Febrero – Junio 2021

Carrera:
Ingeniería en Tecnologías de la Información y Comunicaciones

Materia y serie:
Minería de datos BDD-1703TI9A

Unidad a evaluar: Unidad III

Nombre de la Tarea:
Práctica 5

Nombre del Alumno:
Hernández Negrete Juan Carlos 16212021
Sifuentes Martinez Manuel Javier 17212934

Nombre del docente:
José Christian Romero Hernández

Data visualization analysis in the logistic regression model

Change of directory

The getwd function is used to obtain the current path where Rsearches studio, and it needs to be changed to access the folder where the file to be used is stored. This is why the following function setwd is found, it allows changing the path to the one that the user wants.

```
getwd ()
setwd ("../Desktop/DataMining/MachineLearning/SVM")
getwd ()
```

Import and section dataset

Once the directory has changed, the dataset to be used is loaded into memory, saving it in the dataset variable. The next step is to select the rows and columns to be used, and this is achieved by defining that all the rows will be used and only the last 3 columns with 3: 5.

```
dataset <- read.csv ('Social_Network_Ads.csv')
dataset <- dataset [, 3:5]
```

Encoding the target function as a factor

The purchased column has values of 1 and 0, and what will be done is encoding these characteristics as a factor, in this way the data will be represented as categorical to be able to be graphed.

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

Divide the dataset into a training and test set

The caTools library is loaded to access the sample.split function, which is used to divide the dataset at 2, with a SplitRatio of .75, this means that there will be 100 data for the test set and 300 for the training set. The set.seed function helps to be able to generate a sequence of random numbers, and a specific number is defined so that the results can be replicated.

```
library(caTools)
set.seed (123)
split <- sample.split (dataset $ Purchased, SplitRatio = 0.75)
training_set <- subset (dataset, split == TRUE)
test_set <- subset (dataset, split == FALSE)
```

Scale of characteristics

The scale function is used on column number three of both sets to normalize the stored data, and it is overwritten in their respective dataset. The goal is to improve the predictive accuracy of the algorithm.

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

Adaptation of the logistic regression to the training set

The e1071 library is loaded to be able to use the svm function, from In this way, we can now proceed to fit the SVM classifier data to the training set. The first of its arguments is formula, and it is equal to purchased, since it is what we want to predict, and it is indicated that all its characteristics are taken with all its columns. The data argument is the source from which your data will be taken, as for type, there are different types of arguments, the practice has classification and not regression purposes, so C-classification is specified. Finally there is kernel, in this there are also several options, here the most basic was chosen, but it can be changed to compare which one gives better performance.

```
library(e1071)
classifier = svm (formula = Purchased ~.,
                  data = training_set,
                  type = 'C-classification',
                  kernel = 'linear')
```

Predicting the results of the test set

The following line executes the classifier on the test set and it is specified that the prediction is made on column 3 of the mentioned set.

```
y_pred = predict (classifier, newdata = test_set [-3])
y_pred
```

Making a confusion matrix

To find out how accurate the predictions were, a confusion matrix is used. Adding their false positives and negatives gives the error rate. In the prediction made in the previous step, a 20% error was obtained.

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

	y_pred	
	0	1
0	57	7
1	13	23

Viewing the results of the training set

The following code helps to visualize the existing trends over the training set. The set variable stores the dataset from which the data will be obtained.

Afterwards, the red region and a green region are created through the parameter "by", and the equal value 0.01 is interpreted as a 0 or 1. and from this, it is classified as green or red. The +1 and -1 values result in the space around the edges, this is done so that the points are not too close together. The axes are also defined, specifying the columns of the dataset, Age and Salary. And in order not to show the default names of the table, the colnames () function is used to assign one decided by the user.

The classifier is used to predict the result of each of the pixel bits mentioned above. Everything previously described is graphed with the help of the plot function, column number 3 of the variable stored in set is passed, and two vectors, which are X1 and X2. The main title and that of the “x” and “y” axes are changed with main and xlab and ylab respectively. The result that is obtained is a graph with points but without color.

The contour function creates the boundaries of the plotted values, generating the line between the colors green and red. Finally, there is the points function, in which the values stored in ypred are reviewed and the ifelse function is used to color the displayed points. Running everything together generates a graph with a dispersion of points colored green and red, a red region and a green region divided by a horizontal line.

```
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 = predict (classifier, newdata = grid_set)
plot (set [, -3],
     main = 'SVM (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'))
```

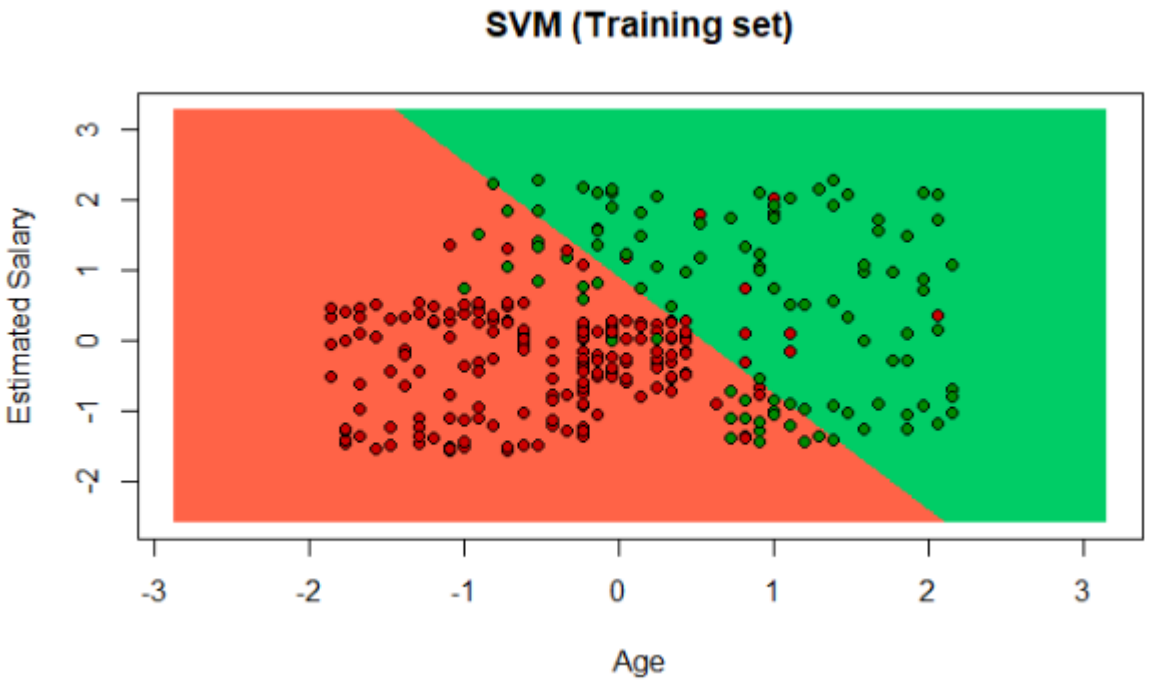
Viewing the test set results

Steps to construct the graph of the data in the test set are the same as shown and explained above, only substituting the source of the data.

```
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 = predict (classifier, newdata = grid_set)
plot (set [, -3], main = 'SVM (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'))
```

Conclusion of training set graph

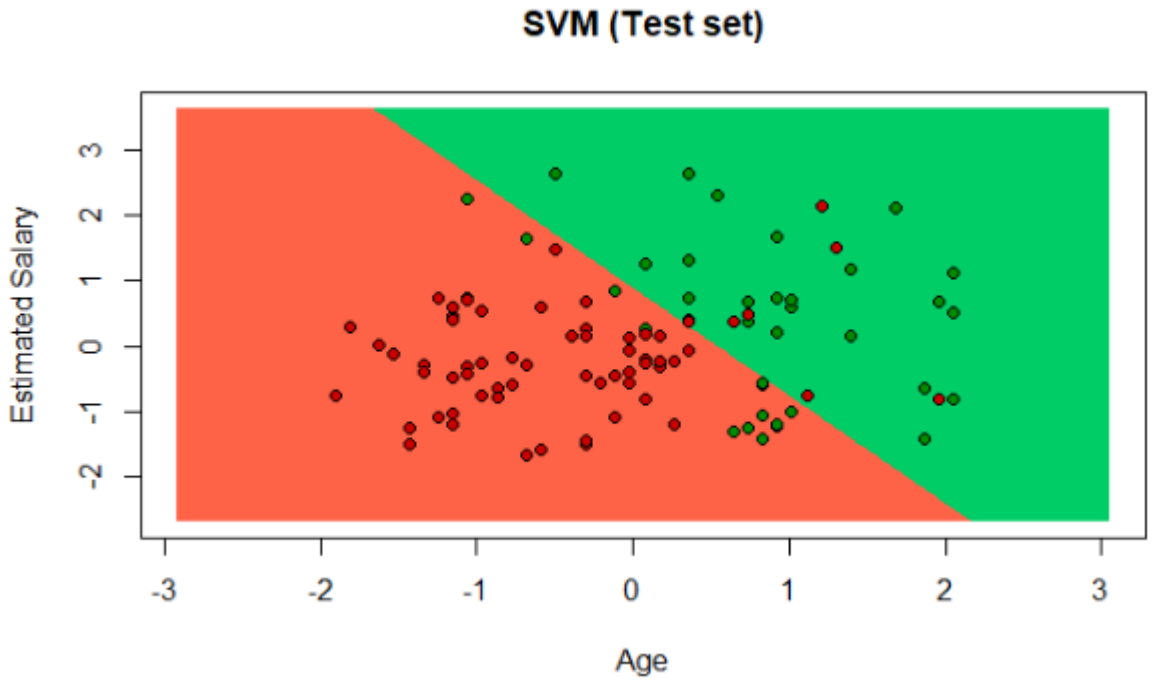
The center line separates the two zones colored, green and red, thus creating the way the data set is classified. The dispersion of the points is divided into two areas of the background colors, and the colors also have two different colors, and can be found in areas of opposite color. The points that do not coincide with the background color are erroneous predictions made in the training set, which would be false positives and false negatives. The red dots indicate to customers that they did not buy an SUV, while the green dots found in the same color region are customers who did buy an SUV.



Graph 1. Training set Graph

Conclusion of the test set

The number of incorrect predictions is similar to the percentage obtained previously, this also considering that it is a smaller number of data. The meaning of the regions and the points shown remains the same, with the points not corresponding to whether their color in their respective region is erroneous predictions, and those that do coincide, the red zone represents the customers who did not buy an SUV and the green those that did.



Graph 2. Test set