**Machine Learning with R**

Support Vector Machines (SVMs) are a set of supervised learning models used for classification and regression analysis. Most often SVM is used for classification. The objective of SVM is to find the most optimal way to separate data according to the defined outputs. SVM does this by identifying the support vectors of the data set. This creates the hyperplane that separate the classes of data.

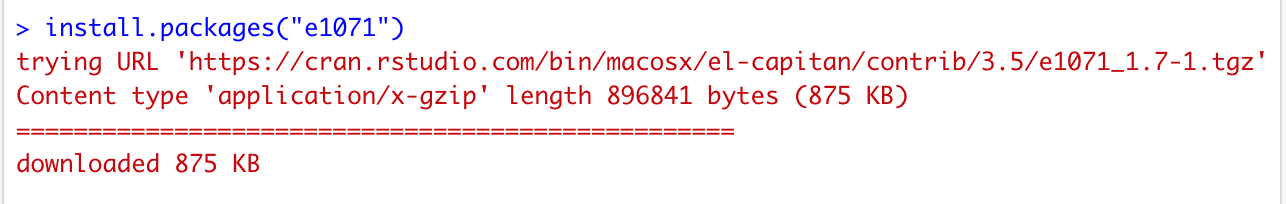
The current assignment is broken into two parts:

Part 1 of the assignment required using SVM for the XOR problem. XOR is a logical operation that returns a true output if one of the input values is true and the other is false. XOR is considered a classification problem since the expected outputs are known in advance.

Part 2 of the assignment required expanding the ideas from the XOR problem to new problems. This section requires changing the XOR problem to an AND problem using SVM. In contrast to XOR, the AND logical operator returns a true output only when both inputs values are true.

**XOR Classification Problem**

1. The first step is to install the e1071 package with the install.package() function.



1. The next step is to load the “e1071” package with the library function and then write the command for SVM. In this step we reset all variables to 0.

A screenshot of a cell phone

Description automatically generated

1. Directly following we create a 4,2 array with its input values. Calling the array “x” provides a truth table containing the results of the created array.

A screenshot of a cell phone

Description automatically generated

* For the truth table, 1 can be represented as True and 0 can be represented as False. For the XOR problem, the output returns True with the inputs (True, False) or (False, True). The output returns False with the inputs (True, True) or (False, False).

1. The fourth step is to create a vector of factors for “y”. The vector of factors contains the outputs for the input values of the truth table. Therefore, 0 XOR 0 = 0 (False), 0 XOR 1 = 1 (True), 1 XOR 0 = 1 (True), 1 XOR 1 = 0 (False).

A screenshot of a cell phone

Description automatically generated

1. The fifth step is to create the SVM model. The SVM model trains the “e1071” package on what the correct answers should be.

A screenshot of a cell phone

Description automatically generated

1. Next, we display information about the created models. The first function summary(model) gets the summary output, while print(model) prints information on the model. The output shows what function and parameters were used to create the model, the number of support vectors, number of classes, and levels.

A screenshot of a cell phone

Description automatically generatedA screenshot of a cell phone

Description automatically generated

1. The final step is to test our original data with the output model. To do this we use the predict function. The predictions provide a list of four x inputs with the predicted outputs. The predicted outputs were the same outputs we trained in “y”, which we anticipated.

A screenshot of a cell phone

Description automatically generated

**AND Problem**

The following steps changes the original XOR problem into an AND problem.

1. Since we already have the “e1071” package installed, all we need to do is load the package and then call the support vector machine. In this step we reset all variables to 0.

A screenshot of a cell phone

Description automatically generated

1. After we develop a 4,2 array with its input values. After we call “x” which provides a truth table with the results of the created array.

A screenshot of a cell phone

Description automatically generated

* For the truth table, 1 can be represented as True and 0 can be represented as False. For the AND problem, the output returns True with the inputs (True, True). The output returns False with the inputs (False, False), (True, False) or (False, True).

1. The third step is to create a vector of factors for “y”. The vector of factors contains the respective outputs from the table from the previous step. Therefore, 0 && 0 = 0 (False), 0 && 1 = 0 (False), 1 && 0 = 1 (False), 1 && 1 = 1 (True).

A close up of a logo

Description automatically generated

1. The fourth step is to create the SVM model. The SVM model trains the “e1071” package on what the correct answers should be.

A screenshot of a cell phone

Description automatically generated

1. The following step is to display some information about the created models. As previously mentioned, the first function summary(model) gets the summary output, while print(model) prints information on the model. The output shows what function and parameters were used to create the model, the number of support vectors, number of classes, and levels.

A screenshot of a cell phone

Description automatically generatedA screenshot of a cell phone

Description automatically generated

1. The final step is to see the SVM predictions. This step tests our original data with the output model. To do this we use the predict function. The predictions provide a list of four x inputs with the predicted outputs. The predicted outputs confirmed the same outputs we trained in “y”.

A screenshot of a cell phone

Description automatically generated