**Tecnológico Nacional De México**

**Instituto Tecnológico De Tijuana**

**Subdirección Académica**

**Departamento de Sistemas y Computación**

**Semestre Enero - Junio 2022**

**Ingeniería Informática**

**Datos Masivos**

**Práctica 5 - Multilayer Perceptron classifier**

**Unidad 2**

**Perez Ortega Victoria Valeria No.18210718**

**Israel López Pablo No.17210585**

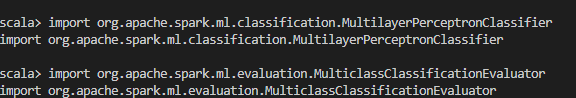
**JOSE CHRISTIAN ROMERO HERNANDEZ**

**Tijuana, B.C. a 18 de Mayo de 2022.**

import org.apache.spark.ml.classification.MultilayerPerceptronClassifier

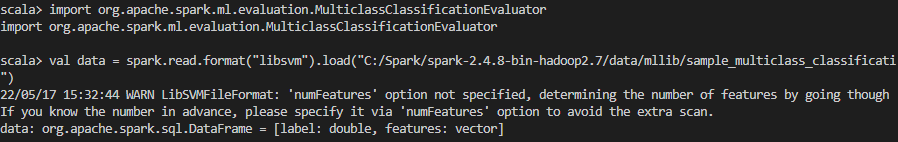
import org.apache.spark.ml.evaluation.MulticlassClassificationEvaluator

// Load the data stored in LIBSVM format as a DataFrame. - Carga los datos almacenados en formato LIBSVM como DataFrame.



//val data = spark.read.format("libsvm").load("data/mllib/sample\_multiclass\_classification\_data.txt")

val data = spark.read.format("libsvm").load("C:/Spark/spark-2.4.8-bin-hadoop2.7/data/mllib/sample\_multiclass\_classification\_data.txt")

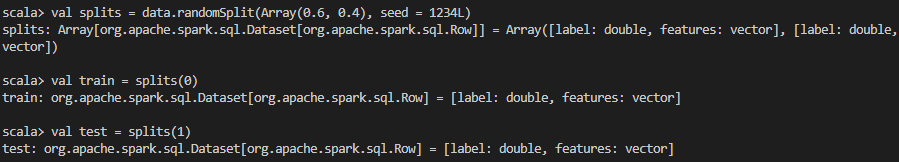


// Split the data into train and test - Divide los datos

val splits = data.randomSplit(Array(0.6, 0.4), seed = 1234L)

val train = splits(0)

val test = splits(1)



// specify layers for the neural network: especificar capas para la red neuronal:

// input layer of size 4 (features), two intermediate of size 5 and 4 capa de entrada de tamano 4 (features), dos intermedias de tamano 5 y 4

// and output of size 3 (classes) y salida de tamano 3 (classes)

val layers = Array[Int](4, 5, 4, 3)



// create the trainer and set its parameters - Crea el trainer y establece sus parametros.

val trainer = new MultilayerPerceptronClassifier()



.setLayers(layers)



.setBlockSize(128)



.setSeed(1234L)



.setMaxIter(100)



// train the model \* entrena el model

val model = trainer.fit(train)

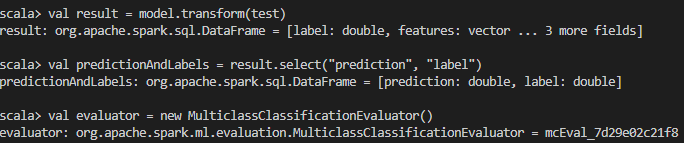


// compute accuracy on the test set precision de - calculo en el conjunto de prueba

val result = model.transform(test)

val predictionAndLabels = result.select("prediction", "label")

val evaluator = new MulticlassClassificationEvaluator()



.setMetricName("accuracy")



println(s"Test set accuracy = ${evaluator.evaluate(predictionAndLabels)}")

