Tecnológico Nacional de México

INSTITUTO TECNOLÓGICO CAMPUS TIJUANA

ING. EN SISTEMAS COMPUTACIONALES

Subdirección Académica

Departamento de Sistemas y Computación

BDD-1704 TI9A - 6:00pm-7:00pm

ASIGNATURA:

Datos Masivos

SEMESTRE:

Septiembre- Enero 2020

Tarea:

Explicación de lo que yo entendí del código:

MultilayerPerceptronClassifier.scala

MAESTRO:

JOSE CHRISTIAN ROMERO HERNANDEZ

30/11/2020

// Se importa MultilayerPerceptronClassifier y MulticlassClassificationEvaluator

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

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

// Se importa la sesión de Spark

import org.apache.spark.sql.SparkSession

// Creacion del objeto MultilayerPerceptronClassifier

object MultilayerPerceptronClassifierExample {

// Se define la funcion main la cual tiene como parámetro un Array de tipo string

def main(): Unit = {

// Se crea el objeto de la clase SparkSession, y a la app se le da el nombre de

// MultilayerPerceptronClassifierExample

val spark = SparkSession.builder.appName("MultilayerPerceptronClassifierExample").getOrCreate()

// $example on$

// Se cargan los datos en formato libsvm del archivo como un DataFrame

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

// Se dividen los datos en entrenamiento y prueba

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

val train = splits(0)

val test = splits(1)

// Se especifican las capas de la red neuronal:

// La capa de entrada es de tamaño 4 (características), dos capas intermedias

// una de tamaño 5 y la otra de tamaño 4

// y 3 de salida (las clases)

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

// Se establecen los parámetros de entrenamiento

val trainer = new MultilayerPerceptronClassifier().setLayers(layers).setBlockSize(128).setSeed(1234L).setMaxIter(100)

// Se entrena el modelo

val model = trainer.fit(train)

// Se calcula la precisión de los datos de prueba

val result = model.transform(test)

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

val evaluator = new MulticlassClassificationEvaluator().setMetricName("accuracy")

// Se imprime la exactitud del modelo

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

spark.stop()

}

}

// scalastyle:on println

//Resultado

