**TECNOLÓGICO​ ​NACIONAL​ ​DE​ ​MÉXICO**

**INSTITUTO TECNOLÓGICO DE TIJUANA**

**SUBDIRECCIÓN ACADÉMICA**

**DEPARTAMENTO DE SISTEMAS Y COMPUTACIÓN**

**SEMESTRE:**Enero - Junio 2020

**CARRERA:**Ing. Tecnologías de la Información y Comunicaciones

**MATERIA:**Datos Masivos

**TÍTULO:​ ​**Decision Tree Classifier Equipo # 1

**UNIDAD​ ​A​ ​EVALUAR:​**​Unidad​ ​2

**NOMBRE Y NÚMERO DE CONTROL DEL ALUMNO:**Diaz Martinez Ruben Emilio # 15210791

**NOMBRE DEL MAESTRO (A):**Dr. Christian Romero Hernandez

import org.apache.spark.ml.Pipeline

import org.apache.spark.ml.classification.DecisionTreeClassificationModel

import org.apache.spark.ml.classification.DecisionTreeClassifier

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

import org.apache.spark.ml.feature.{IndexToString, StringIndexer, VectorIndexer}

import org.apache.spark.sql.SparkSession

def main(): Unit = {

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

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

val labelIndexer = new StringIndexer().setInputCol("label").setOutputCol("indexedLabel").fit(data)

val featureIndexer = new VectorIndexer().setInputCol("features").setOutputCol("indexedFeatures").setMaxCategories(4).fit(data)

val Array(trainingData, testData) = data.randomSplit(Array(0.7, 0.3))

val dt = new DecisionTreeClassifier().setLabelCol("indexedLabel").setFeaturesCol("indexedFeatures")

val labelConverter = new IndexToString().setInputCol("prediction").setOutputCol("predictedLabel").setLabels(labelIndexer.labels)

val pipeline = new Pipeline().setStages(Array(labelIndexer, featureIndexer, dt, labelConverter))

val model = pipeline.fit(trainingData)

val predictions = model.transform(testData)

predictions.select("predictedLabel", "label", "features").show(5)

val evaluator = new MulticlassClassificationEvaluator().setLabelCol("indexedLabel").setPredictionCol("prediction").setMetricName("accuracy")

val accuracy = evaluator.evaluate(predictions)

println(s"Test Error = ${(1.0 - accuracy)}")

val treeModel = model.stages(2).asInstanceOf[DecisionTreeClassificationModel]

println(s"Learned classification tree model:\n ${treeModel.toDebugString}")

spark.stop()

}

main()