**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:​ ​**Gradient Boosted Tree Equipo # 3

**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.{GBTClassificationModel, GBTClassifier}

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

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

import org.apache.spark.sql.SparkSession

import org.apache.log4j.\_

Logger.getLogger("org").setLevel(Level.ERROR)

def main(): Unit = {

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

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

data.printSchema()

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 gbt = new GBTClassifier().setLabelCol("indexedLabel").setFeaturesCol("indexedFeatures").setMaxIter(10).setFeatureSubsetStrategy("auto")

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

val pipeline = new Pipeline().setStages(Array(labelIndexer, featureIndexer, gbt, 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 gbtModel = model.stages(2).asInstanceOf[GBTClassificationModel]

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

spark.stop()

}

main()