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:

GradientBoostingClassificationExample.scala

MAESTRO:

JOSE CHRISTIAN ROMERO HERNANDEZ

26/11/2020

//importamos las librerias

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}

// cargamos y analizamos el archivo data, lo comvertimos a dataframe

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

// Index labels, agregamos el metadato a la columna label.

// Ajustamos todo el conjunto de datos para incluir todos los labels a index.

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

// Identifica automáticamente las características categóricas e indexalas.

// Establezca maxCategories para que las entidades con un valores distintos > 4 se traten como continuas.

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

// Divida los datos en conjuntos de prueba y entrenamiento (el 30% se reserva para probar).

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

// entrenando al modelo GBT

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

// convirtiendo indexed labels devuelta al labels original.

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

// cadena de indexers y GBT en Pipeline.

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

// entrenando el modelo. Esto también ejecuta el indexers.

val model = pipeline.fit(trainingData)

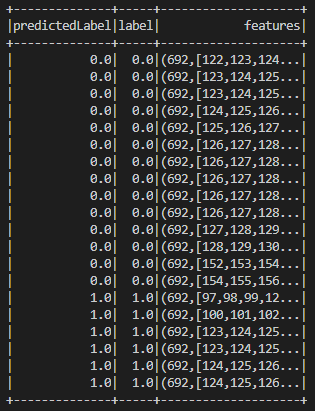
// haciendo predicciones

val predictions = model.transform(testData)

// Seleccionamos las filas de ejemplo para mostrar.

predictions.select("predictedLabel", "label", "features").show(20)

//Resultados



// Select (prediction, true label) and compute test error.

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

val accuracy = evaluator.evaluate(predictions)

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

//Resultado



val gbtModel = model.stages(2).asInstanceOf[GBTClassificationModel]

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

Resultados:

