**Tecnológico Nacional De México**

**Instituto Tecnológico De Tijuana**

**Subdirección Académica**

**Departamento de Sistemas y Computación**

**Semestre Enero - Junio 2022**

**Ingeniería Informática**

**Datos Masivos**

**Práctica 1 - Conceptos básicos de estadística**

**Unidad 2**

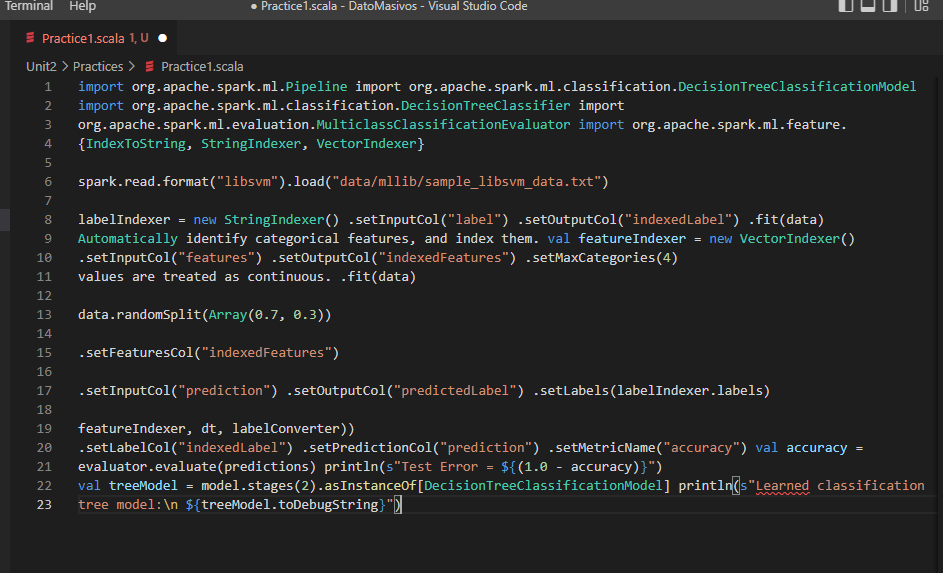
**Perez Ortega Victoria Valeria No.18210718**

**Israel López Pablo No.17210585**

**JOSE CHRISTIAN ROMERO HERNANDEZ**

**Tijuana, B.C. a 04 de Mayo de 2022.**

Documentar y ejecutar el ejemplo de la documentación de spark de **Basic Statistics** , en su branch correspondiente.



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}

spark.read.format("libsvm").load("data/mllib/sample\_libsvm\_data.txt")

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

Automatically identify categorical features, and index them. val featureIndexer = new VectorIndexer()

.setInputCol("features") .setOutputCol("indexedFeatures") .setMaxCategories(4)

values are treated as continuous. .fit(data)

data.randomSplit(Array(0.7, 0.3))

.setFeaturesCol("indexedFeatures")

.setInputCol("prediction") .setOutputCol("predictedLabel") .setLabels(labelIndexer.labels)

featureIndexer, dt, labelConverter))

.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}")