TECNOLÓGICO NACIONAL DE MÉXICO

INSTITUTO TECNOLÓGICO DE TIJUANA
SUBDIRECCIÓN ACADÉMICA
DEPARTAMENTO DE SISTEMAS Y COMPUTACIÓN

SEMESTRE:

Enero - Junio 2022

CARRERA:

Ing. en Sistemas Computacionales

MATERIA:

Datos Masivos

TÍTULO ACTIVIDAD:

Practica 2

NOMBRE Y NÚMERO DE CONTROL DEL ALUMNO:

Hernandez Pablo Anahi Del Carmen – 18210486
Real Castro Manuel Angel – 18212253

Here we start Spark and add the libraries that we were going to use

```
scala> import org.apache.spark.ml.Pipeline
import org.apache.spark.ml.Pipeline

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

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

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

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

In this part we define all the variables and set the pipeline

```
scala> val labelIndexer = new StringIndexer().setInputCol("label").setOutputCol("indexedLabel").fit(data)
labelIndexer: org.apache.spark.ml.feature.StringIndexerModel = strIdx_6ce420d983a0

scala> val featureIndexer = new VectorIndexer().setInputCol("features").setOutputCol("indexedFeatures").setMaxCategories
(4).fit(data)
featureIndexer: org.apache.spark.ml.feature.VectorIndexerModel = vecIdx_431d3e777725

scala> val Array(trainingData, testData) = data.randomSplit(Array(0.7, 0.3))
trainingData: org.apache.spark.sql.Dataset[org.apache.spark.sql.Row] = [label: double, features: vector]
testData: org.apache.spark.sql.Dataset[org.apache.spark.sql.Row] = [label: double, features: vector]

scala> val dt = new DecisionTreeClassifier().setLabelCol("indexedLabel").setFeaturesCol("indexedFeatures")
dt: org.apache.spark.ml.classification.DecisionTreeClassifier = dtc_201a4f940442

scala> val labelConverter = new IndexToString().setInputCol("prediction").setOutputCol("predictedLabel").setLabels(label
Indexer.labels)
labelConverter: org.apache.spark.ml.feature.IndexToString = idxToStr_db888892b836

scala> val pipeline = new Pipeline().setStages(Array(labelIndexer, featureIndexer, dt, labelConverter))
pipeline: org.apache.spark.ml.Pipeline = pipeline_c8db303ca75a
```

Here we train de model, get the predictions to show it in the end.

```
scala> val model = pipeline.fit(trainingData)
model: org.apache.spark.ml.PipelineModel = pipeline_c8db303ca75a

scala> val predictions = model.transform(testData)
predictions: org.apache.spark.sql.DataFrame = [label: double, features: vector ... 6 more fields]

scala> predictions.select("predictedLabel", "label", "features").show(5)
+--------------+-----+--------------------+
|predictedLabel|label|            features|
+--------------+-----+--------------------+
|           0.0|  0.0|(692,[98,99,100,1...|
|           0.0|  0.0|(692,[121,122,123...|
|           0.0|  0.0|(692,[122,123,148...|
|           0.0|  0.0|(692,[124,125,126...|
|           0.0|  0.0|(692,[124,125,126...|
+--------------+-----+--------------------+
only showing top 5 rows
```

Here we print the metrics of the classifier.

```
scala> val accuracy = evaluator.evaluate(predictions)
accuracy: Double = 1.0

scala> println(s"Test Error = ${(1.0 - accuracy)}")
Test Error = 0.0

scala> val treeModel = model.stages(2).asInstanceOf[DecisionTreeClassificationModel]
treeModel: org.apache.spark.ml.classification.DecisionTreeClassificationModel = DecisionTreeClassificationModel (uid=dt
_201a4f940442) of depth 2 with 5 nodes

scala> println(s"Learned classification tree model:\n ${treeModel.toDebugString}")
Learned classification tree model:
 DecisionTreeClassificationModel (uid=dtc_201a4f940442) of depth 2 with 5 nodes
  If (feature 434 <= 70.5)
   If (feature 99 in {2.0})
    Predict: 0.0
   Else (feature 99 not in {2.0})
    Predict: 1.0
  Else (feature 434 > 70.5)
   Predict: 0.0
```