

TECNOLÓGICO NACIONAL DE MÉXICO
INSTITUTO TECNOLÓGICO DE TIJUANA
SUBDIRECCIÓN ACADÉMICA
DEPARTAMENTO DE SISTEMAS Y COMPUTACIÓN

SEMESTRE:

Enero - Junio 2021

CARRERA:

Ingeniería en Sistemas Computacionales

MATERIA:

Datos Masivos

TÍTULO:

Practica-Decision Tree Classifier

UNIDAD A EVALUAR:

Unidad -2

ALUMNO: JUAN ANTONIO ACEVES ZAMORA

NO. CONTROL:16210502

NOMBRE DEL DOCENTE :

JOSE CHRISTIAN ROMERO HERNANDEZ

Decision Tree Classifier

Importar las librerías necesarias.

```
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.Session
```

Cargue los datos almacenados en formato LIBSVM como un DataFrame.

```
val data =
spark.read.format("libsvm").load("/home/eduardo/Escritorio/expo/sa
mple_libsvm_data.txt")
```

Etiquetas de índice, agregando metadatos a la columna de etiquetas. Encajar en el conjunto de datos completo para incluir todas las etiquetas en el índice.

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

Identificar automáticamente características categóricas e indexar.

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

Divida los datos en conjuntos de prueba y entrenamiento (30% reservado para pruebas).

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

Entrene un modelo Decision Tree.

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

Convertir etiquetas indexadas de nuevo a etiquetas originales.

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

Cadena de indexadores y árbol en un Pipeline.

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

Modelo de train. Esto también ejecuta a los indexadores.

```
val model = pipeline.fit(trainingData)
```

Hacer predicciones.

```
val predictions = model.transform(testData)
```

Seleccione filas de ejemplo para mostrar. En este caso solo serán 5

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

Seleccione (predicción, etiqueta verdadera).

```
val evaluator = new  
MulticlassClassificationEvaluator().setLabelCol("indexedLabel").set  
tPredictionCol("prediction").setMetricName("accuracy")
```

Calcule el error de prueba.

```
val accuracy = evaluator.evaluate(predictions)  
println(s"Test Error = ${1.0 - accuracy}")
```

Mostrar por etapas la clasificación del modelo de árbol

```
val treeModel =
```

```
model.stages(2).asInstanceOf[DecisionTreeClassificationModel]  
println(s"Learned classification tree model:\n  
${treeModel.toDebugString}")
```