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:

Práctica-Random Forest Classifier

UNIDAD A EVALUAR:

Unidad -2

ALUMNO: JUAN ANTONIO ACEVES ZAMORA

NO. CONTROL:16210502

NOMBRE DEL DOCENTE :

JOSE CHRISTIAN ROMERO HERNANDEZ

Random Forest Classifier

Importamos las bibliotecas necesarias para este ejemplo que realizaron nuestros compañeros.

```
import org.apache.spark.ml.Pipeline
import
org.apache.spark.ml.classification.{RandomForestClassificationMode
l, RandomForestClassifier}
import
org.apache.spark.ml.evaluation.MulticlassClassificationEvaluator
import org.apache.spark.ml.feature.{IndexToString, StringIndexer,
VectorIndexer}
import org.apache.spark.sql.SparkSession
```

Comenzamos una sesión de Spark

```
val spark = SparkSession.builder().getOrCreate()
```

Cargamos los datos y convertimos a DataFrame

```
val data =
spark.read.format("libsvm").load("data/mllib/sample_libsvm_data.tx
t")
```

Creamos dos objetos en los que introducimos el nombre de los valores de entrada y el nombre de la salida

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

Dividimos los datos en conjuntos de entrenamiento y pruebas (70% de entrenamiento y 30% para pruebas)

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

Creamos la variable rf, cargamos los datos y los valores de las columnas para el modelo Bosque aleatorio

```
val rf = new
RandomForestClassifier().setLabelCol("indexedLabel").setFeaturesCo
l("indexedFeatures").setNumTrees(10)
```

Convierta las etiquetas indizados en etiquetas originales

```
val labelConverter = new
IndexToString().setInputCol("prediction").setOutputCol("predictedL
abel").setLabels(labelIndexer.setLabels)
```

Ajustamos el pipeline con indexadores de cadena y bosque

```
val pipeline = new Pipeline().setStages(Array(labelIndexer,
featureIndexer, rf, labelConverter))
val model = pipeline.fit(trainingData)
```

Creamos la variable donde se harán las predicciones y mostraremos filas de prueba

```
val predictions = model.transform(testData)
predictions.select("predictedLabel", "label", "features").show(5)
```

Creamos un objeto donde se llevará a cabo la evaluación, creamos la variable de precisión donde se llevará a cabo la evaluación y la predicción

```
val evaluator = new
MulticlassClassificationEvaluator().setLabelCol("indexedLabel").se
tPredictionCol("prediction").setMetricName("accuracy")
val accuracy = evaluator.evaluate(predictions)
println(s"Test Error = ${(1.0 - accuracy)}")
```

Finalmente imprimimos Random Forest Model

```
val rfModel =
model.stages(2).asInstanceOf[RandomForestClassificationModel]
println(s"Learned classification forest model:\n
${rfModel.toDebugString}")
```