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-Gradient Boosted Tree Classifier

UNIDAD A EVALUAR:

Unidad -2

ALUMNO: JUAN ANTONIO ACEVES ZAMORA

NO. CONTROL:16210502

NOMBRE DEL DOCENTE :

JOSE CHRISTIAN ROMERO HERNANDEZ

Importamos las bibliotecas que ocupamos...

```
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 el archivo txt de la ruta establecida

```
val data =
spark.read.format("libsvm").load("sample_libsvm_data.txt")
```

Crearemos una columna usando stringIndexer para que los datos tengan su categorización

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

Creamos un vector que tendrá un máximo de 4 categorías

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

Separamos los datos en dos partes, una llamada entrenamiento con un 70% y la otra prueba con un 30%

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

El modelo GPT está entrenado

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

Convertimos las etiquetas indexadas a etiquetas originales

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

Ajustamos Pipeline y los índices

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

Se entrena el modelo y se ejecutan los indexadores

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

Creamos las predicciones

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

Seleccionamos las 5 primeras filas para mostrarlas

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

Seleccionamos predicción y cálculo del error de prueba

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

Finalmente mostrar lo aprendido del modelo GBT

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