



**EDUCACIÓN**  
SECRETARÍA DE EDUCACIÓN PÚBLICA



TECNOLÓGICO  
NACIONAL DE MÉXICO®

TECNOLÓGICO NACIONAL DE MÉXICO

INSTITUTO TECNOLÓGICO DE TIJUANA

SUBDIRECCIÓN ACADÉMICA

DEPARTAMENTO DE SISTEMAS Y COMPUTACIÓN

NOMBRE DE LOS ALUMNOS:

RAYMUNDO HIRALES LAZARENO (N. CONTROL: 17212339)

PAULA ANDREA RAMOS VERDIN (N. CONTROL: 18210721)

Carrera: Ingeniería Informática

MATERIA: Datos Masivos

PROFESOR: JOSE CHRISTIAN ROMERO HERNANDEZ

Practica 5

FECHA: 18/05/22

```
import org.apache.spark.ml.classification.NaiveBayes import
org.apache.spark.ml.evaluation.MulticlassClassificationEvaluator import org.apache.spark.sql.SparkSession

//Cargar los datos especificando la ruta del archivo

val data = spark.read.format("libsvm").load("C:/spark/spark-2.4.8-bin-
hadoop2.7/data/mllib/sample_libsvm_data.txt")

println ("Numero de lineas en el archivo de datos:" + data.count ())

//Mostrar las primeras 20 líneas por defecto

data.show()

//Divida aleatoriamente el conjunto de datos en conjunto de entrenamiento y conjunto de prueba de acuerdo
con los pesos proporcionados. También puede especificar una seed

val Array (trainingData, testData) = data.randomSplit (Array (0.7, 0.3), 100L) // El resultado es el tipo de la
matriz, y la matriz almacena los datos de tipo DataSet

//Incorporar al conjunto de entrenamiento (operación de ajuste) para entrenar un modelo bayesiano val
naiveBayesModel = new NaiveBayes().fit(trainingData)

//El modelo llama a transform() para hacer predicciones y generar un nuevo DataFrame.

val predictions = naiveBayesModel.transform(testData)

//Salida de datos de resultados de predicción predictions.show()

//Evaluación de la precisión del modelo

val evaluator = new
MulticlassClassificationEvaluator().setLabelCol("label").setPredictionCol("prediction").setMetricName("accuracy
") // Precisión val precision = evaluator.evaluate (predictions)

//Imprimir la tasa de error println ("tasa de error =" + (1-precision))

//Resultado
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
scala> predictions.show()
22/05/17 01:44:34 WARN BLAS: Failed to load implementation from: com.github.fommil.netlib.NativeSystemBLAS
22/05/17 01:44:34 WARN BLAS: Failed to load implementation from: com.github.fommil.netlib.NativeRefBLAS
+-----+-----+-----+-----+-----+
|label|      features|      rawPrediction|probability|prediction|
+-----+-----+-----+-----+-----+
| 0.0|(692,[122,123,124...|[-190595.07825499...| [1.0,0.0]|      0.0|
| 0.0|(692,[123,124,125...|[-246607.82713076...| [1.0,0.0]|      0.0|
| 0.0|(692,[123,124,125...|[-199533.44171742...| [1.0,0.0]|      0.0|
| 0.0|(692,[124,125,126...|[-275837.93657958...| [1.0,0.0]|      0.0|
| 0.0|(692,[125,126,127...|[-258745.16054128...| [1.0,0.0]|      0.0|
| 0.0|(692,[126,127,128...|[-279335.13066876...| [1.0,0.0]|      0.0|
| 0.0|(692,[126,127,128...|[-136866.59032154...| [1.0,0.0]|      0.0|
| 0.0|(692,[126,127,128...|[-208662.62448642...| [1.0,0.0]|      0.0|
| 0.0|(692,[127,128,129...|[-211024.47289349...| [1.0,0.0]|      0.0|
| 0.0|(692,[127,128,129...|[-182940.32083349...| [1.0,0.0]|      0.0|
| 0.0|(692,[152,153,154...|[-93356.564312516...| [1.0,0.0]|      0.0|
| 0.0|(692,[153,154,155...|[-260165.09822408...| [1.0,0.0]|      0.0|
| 0.0|(692,[153,154,155...|[-207398.84940196...| [1.0,0.0]|      0.0|
| 0.0|(692,[155,156,180...|[-229364.82012475...| [1.0,0.0]|      0.0|
| 0.0|(692,[234,235,237...|[-95021.471295301...| [1.0,0.0]|      0.0|
| 1.0|(692,[100,101,102...|[-143311.69653049...| [0.0,1.0]|      1.0|
| 1.0|(692,[123,124,125...|[-99743.193683546...| [0.0,1.0]|      1.0|
| 1.0|(692,[124,125,126...|[-127676.82708555...| [0.0,1.0]|      1.0|
| 1.0|(692,[125,126,127...|[-102406.94970386...| [0.0,1.0]|      1.0|
| 1.0|(692,[125,126,153...|[-81780.123925676...| [0.0,1.0]|      1.0|
+-----+-----+-----+-----+-----+
only showing top 20 rows

scala> val evaluator = new MulticlassClassificationEvaluator().setLabelCol("label").setPredictionCol("prediction").setMetricName("accuracy")
evaluator: org.apache.spark.ml.evaluation.MulticlassClassificationEvaluator = mcEval_ee2417604e2b

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

scala> println ("tasa de error =" + (1-precision))
tasa de error =0.0

scala> 
```