



**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:

GALAVIZ LONA OSCAR EDUARDO (N.CONTROL: 17212993)

MARQUEZ MILLAN SEASHELL VANESSA (N.CONTROL: )

Carrera: Ingeniería Informática

Semestre: 9no

MATERIA: Datos Masivos

PROFESOR: JOSE CHRISTIAN ROMERO HERNANDEZ

Practica 7

Unidad 2

## Development

This is the development of the practice 7 naive bayes, this is a classification we are use to make this practice we need to import two libraries for this, one of this is naivebayes and the other is multiclass

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

We have to load de dataframa as archive text or csv

```
// Load the data stored in LIBSVM format as a DataFrame.
val data = spark.read.format("libsvm").load("sample.txt")
<p>

</p>
we make a split the dataframe in two parts
```scala
// Split the data into training and test sets (30% held out for testing)
val Array(trainingData, testData) = data.randomSplit(Array(0.7, 0.3), seed =
1234L)
```

```
scala> val Array(trainingData, testData) = data.randomSplit(Array(0.7, 0.3)
, seed = 1234L)
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]
```

in this part we implement the model of naivebayes for a new dataframe

```
// Train a NaiveBayes model.
val model = new NaiveBayes().fit(trainingData)
```

```
scala> val model = new NaiveBayes().fit(trainingData)
[Stage 1:> (0 + 1)
model: org.apache.spark.ml.classification.NaiveBayesModel = NaiveBayesModel
(uid=nb_eb6a6837e653) with 2 classes
```

we execute the model and show the model with peditions

```
// Select example rows to display.
val predictions = model.transform(testData)
predictions.show()
```

```

usuario@ubuntu-20: ~
model: org.apache.spark.ml.classification.NaiveBayesModel = NaiveBayesModel (uid=nb_eb6a6837e653) with 2 classes

scala> val predictions = model.transform(testData)
predictions: org.apache.spark.sql.DataFrame = [label: double, features: vector ... 3 more fields]

scala> predictions.show()
21/11/06 04:52:30 WARN BLAS: Failed to load implementation from: com.github.fommil.netlib.NativeSystemBLAS
21/11/06 04:52:30 WARN BLAS: Failed to load implementation from: com.github.fommil.netlib.NativeRefBLAS
+-----+-----+-----+-----+-----+
|label|      features|rawPrediction|probability|prediction|
+-----+-----+-----+-----+
| 0.0|(692,[95,96,97,12...|[-173678.60946628...|[1.0,0.0]|    0.0|
| 0.0|(692,[98,99,100,1...|[-178107.24302988...|[1.0,0.0]|    0.0|
| 0.0|(692,[100,101,102...|[-100020.80519087...|[1.0,0.0]|    0.0|
| 0.0|(692,[124,125,126...|[-183521.85526462...|[1.0,0.0]|    0.0|
| 0.0|(692,[127,128,129...|[-183004.12461660...|[1.0,0.0]|    0.0|
| 0.0|(692,[128,129,130...|[-246722.96394714...|[1.0,0.0]|    0.0|
| 0.0|(692,[152,153,154...|[-208696.01108598...|[1.0,0.0]|    0.0|
| 0.0|(692,[153,154,155...|[-261509.59951302...|[1.0,0.0]|    0.0|
| 0.0|(692,[154,155,156...|[-217654.71748256...|[1.0,0.0]|    0.0|
| 0.0|(692,[181,182,183...|[-155287.07585335...|[1.0,0.0]|    0.0|
| 1.0|(692,[99,100,101,...|[-145981.83877498...|[0.0,1.0]|    1.0|
| 1.0|(692,[100,101,102...|[-147685.13694275...|[0.0,1.0]|    1.0|
| 1.0|(692,[123,124,125...|[-139521.98499849...|[0.0,1.0]|    1.0|
| 1.0|(692,[124,125,126...|[-129375.46702012...|[0.0,1.0]|    1.0|
| 1.0|(692,[126,127,128...|[-145809.08230799...|[0.0,1.0]|    1.0|
| 1.0|(692,[127,128,129...|[-132670.15737290...|[0.0,1.0]|    1.0|
| 1.0|(692,[128,129,130...|[-100206.72054749...|[0.0,1.0]|    1.0|
| 1.0|(692,[129,130,131...|[-129639.09694930...|[0.0,1.0]|    1.0|
| 1.0|(692,[129,130,131...|[-143628.65574273...|[0.0,1.0]|    1.0|
| 1.0|(692,[129,130,131...|[-129238.74023248...|[0.0,1.0]|    1.0|
+-----+-----+-----+-----+
only showing top 20 rows

```

Finally we show the accuracy they have model of naivebayes

```

// Select (prediction, true label) and compute test error
val evaluator = new
MulticlassClassificationEvaluator().setLabelCol("label").setPredictionCol("prediction").setMetricName("accuracy")
val accuracy = evaluator.evaluate(predictions)
println(s"Test set accuracy = $accuracy")

```

```

scala> val evaluator = new MulticlassClassificationEvaluator().setLabelCol("label").setPredictionCol("prediction").setMetricName("accuracy")
evaluator: org.apache.spark.ml.evaluation.MulticlassClassificationEvaluator = mcEval_422970a89a9a

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

scala> println(s"Test set accuracy = $accuracy")
Test set accuracy = 1.0

scala>

```