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>
<img alt="Logo" src="./../Media/practicenaives.PNG" >
</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] = [lab
el: 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

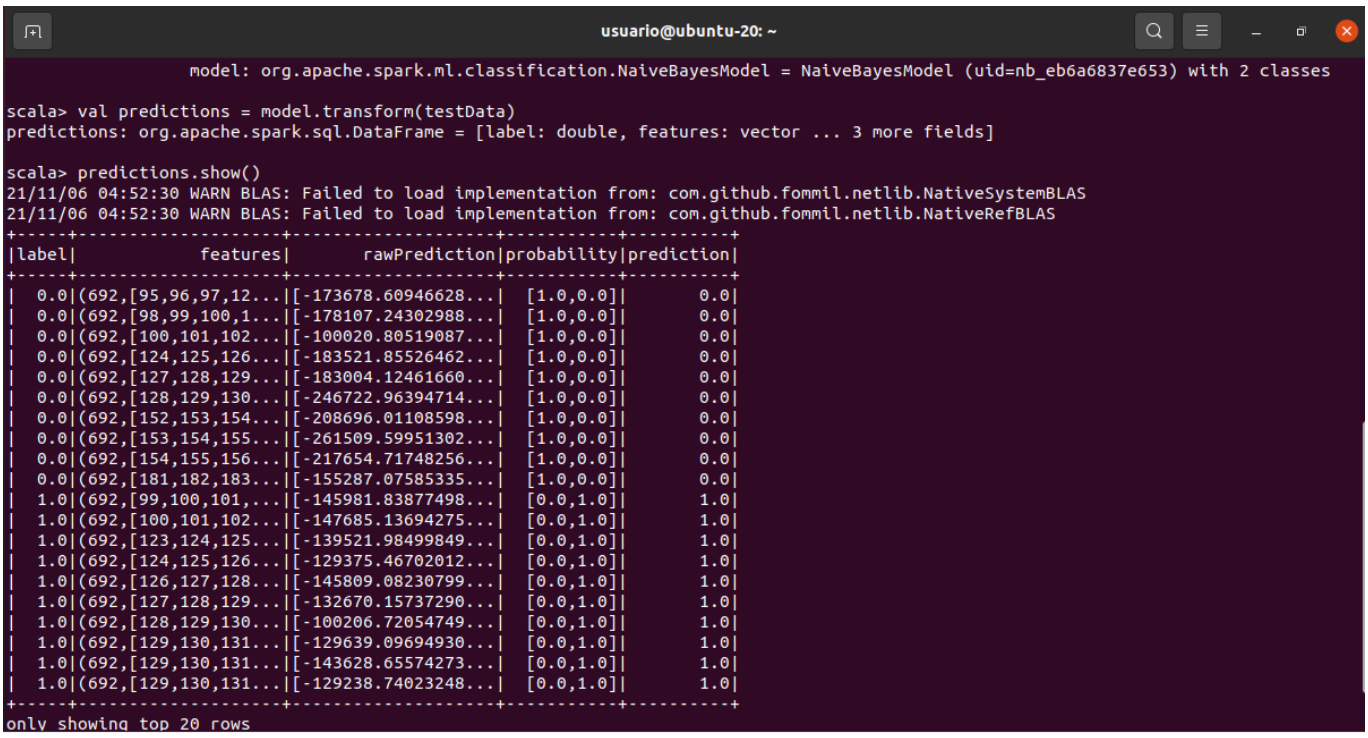
```
// Train a NaiveBayes model.
val model = new NaiveBayes().fit(trainingData)
```

```
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 peditictions

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



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("predict
ion").setMetricName("accuracy")
val accuracy = evaluator.evaluate(predictions)
println(s"Test set accuracy = $accuracy")
```