



EDUCACIÓN
SECRETARÍA DE EDUCACIÓN PÚBLICA



TECNOLÓGICO
NACIONAL DE MÉXICO



**TECNOLÓGICO NACIONAL DE MÉXICO INSTITUTO
TECNOLÓGICO DE TIJUANA**

SUBDIRECCIÓN ACADÉMICA

DEPARTAMENTO DE SISTEMAS Y COMPUTACIÓN

SEMESTRE FEBRERO-JUNIO 2022

CARRERA

Ingeniería en informática

MATERIA

Datos masivos

TÍTULO

Práctica#7

Integrantes:

Munguía silva Edgar Geovanny #17212344

Alicia Guadalupe Pérez López#18210514

NOMBRE DEL MAESTRO

Jose Christian Romero Hernadez

Tijuana Baja California 17 de Mayo del 2022

¿Qué es Naive Bayes?

Es un clasificador probabilístico fundamentado en el teorema de Bayes y algunas hipótesis simplificadoras adicionales.

El modelo se denomina naïve porque trata todas las variables de predicción propuestas como independientes unas de otras. El bayesiano ingenuo es un algoritmo rápido y escalable que calcula las probabilidades condicionales para las combinaciones de atributos y el atributo de objetivo.

Pasos para llevar a cabo el algoritmo.

- El conjunto de datos en una tabla de frecuencias.
- Crear una tabla de probabilidad calculando las correspondientes a que ocurran los diversos eventos.
- La ecuación Naive Bayes se usa para calcular la probabilidad posterior de cada clase.
- La clase con la probabilidad posterior más alta es el resultado de la predicción.

Fórmula de algoritmo con el significado de sus variables.

$$P(A|R) = \frac{P(R|A)P(A)}{P(R)}$$

P(A): Probabilidad de A

P(R|A): Probabilidad de que se de R dado A

P(R): Probabilidad de R

P(A|R): Probabilidad posterior de que se de A dado R

Puntos Fuertes.

- Una manera fácil y rápida de predecir clases, para problemas de clasificación binarios y multiclase.
- El algoritmo se comporta mejor que otros modelos de clasificación, incluso con menos datos de entrenamiento.
- El desacoplamiento de las distribuciones de características condicionales de clase significan que cada distribución puede ser estimada independientemente como si tuviera una sola dimensión.

Puntos débiles.

- Los algoritmos Naive Bayes son conocidos por ser pobres estimadores. Por ello, no se deben tomar muy en serio las probabilidades que se obtienen.
- La presunción de independencia Naive muy probablemente no reflejará cómo son los datos en el mundo real.
- Cuando el conjunto de datos de prueba tiene una característica que no ha sido observada en el conjunto de entrenamiento, el modelo le asignará una probabilidad de cero y será inútil realizar predicciones.

Referencias bibliográficas.

Roman, V. (2021, 9 diciembre). Algoritmos Naive Bayes: Fundamentos e Implementación. Medium.

<https://medium.com/datos-y-ciencia/algoritmos-naive-bayes-fudamentos-e-implemen>

taci%C3%B3n-4bcb24b307f

Gonzalez, L. (2020, 21 agosto). Naive Bayes – Teoría. 🤖 Aprende IA. Recuperado 11 de mayo de 2022, de <https://aprendeia.com/naive-bayes-teoria-machine-learning/>

Cardellino, F. (2021, 28 abril). Cómo funcionan los clasificadores Naive Bayes: con ejemplos de código de Python. freeCodeCamp.org. Recuperado 11 de mayo de 2022, de <https://www.freecodecamp.org/espanol/news/como-funcionan-los-clasificadores-naive-bayes-con-ejemplos-de-codigo-de-python/>

Gonzalez, A. C. L. (2019, 20 septiembre). NAIVE BAYES - TEORÍA | #46 Curso Machine Learning con Python [Vídeo]. YouTube. <https://www.youtube.com/watch?v=949tYJgRvRg>

Ejemplo en código.

```
//Importar las librerías necesarias

import org.apache.spark.ml.classification.NaiveBayes
import
org.apache.spark.ml.evaluation.MulticlassClassificationEvaluator
import org.apache.spark.sql.Session

//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 líneas 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").setPredic
tionCol("prediction").setMetricName("accuracy")
// Precisión
val precision = evaluator.evaluate (predictions)

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

Resultados.

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> []
```