Reportes Científicos de la FACEN

Artículo Original

Reportes científicos de la FACEN, Enero - Junio 2022, 13(1):73–84

http://doi.org/10.18004/rcfacen.2022.13.1.73

Evaluación de exactitud de Naive Bayes y Regresión Logística para clasificación con atributos y clases binarios

Accuracy evaluation of Naive Bayes and Logistic Regression for classification with binary attributes and classes

Edgar López Pezoa[1,4,*], Antoliano Cáceres Estigarribia[2], Sebastián Alberto Grillo[1,5] & Edher Herrera[3,4]

[1]Universidad Nacional de Asunción, Facultad de Ciencias Exactas y Naturales, Departamento de Matemática, San Lorenzo, Paraguay.
[2]Universidad Autónoma de Asunción, Facultad de Ciencias y Tecnologías, Asunción, Paraguay.
[3]Universidad Nacional de Asunción, Facultad de Ciencias Exactas y Naturales, Laboratorio de Análisis Molecular y Elemental, San Lorenzo, Paraguay.
[4] Universidad Nacional de Asunción, Facultad Politécnica, Departamento de Ciencias Básicas, San Lorenzo, Paraguay.
[5] Universidad Nacional de Asunción, Facultad Politécnica, Núcleo de Investigación y Desarrollo Tecnológico (NIDTEC), San Lorenzo, Paraguay.
*Autor correspondiente: epezoa@facen.una.py.

Resumen: En ciencia de los datos, la mayoría de los modelos de clasificación están en la categoría de modelos discriminativos o de modelos generativos. Los modelos discriminativos solamente capturan la relación entre los atributos de una instancia y su clase, mientras que los modelos generativos buscan representar toda la distribución de datos. Aunque la mayoría de los modelos de clasificación sean discriminativos, no se puede asegurar que este tipo de modelos sea mejor que los modelos generativos. En ese sentido, se aborda la comparación de los algoritmos Naive Bayes y Regresión Logística como modelos muy representativos de los clasificadores discriminativos y generativos, respectivamente. En este trabajo son evaluadas la exactitud de los modelos de Naive Bayes y Regresión Logística en función al número de atributos e instancias de un conjunto de datos artificiales, donde tanto los atributos como las clases son binarios. A diferencia de otras metodologías que emplean los conjuntos de datos para aproximar el error de clasificación, este trabajo solo emplea los conjuntos de datos para realizar el entrenamiento de los modelos, mientras que el error de clasificación es calculado de forma exacta para la distribución de los datos. Los experimentos muestran una exactitud de clasificación binaria que tiende a ser levemente mejor para la Regresión Logística usando 50 a 500 instancias de entrenamiento, cuando promediamos los resultados de distribuciones generadas aleatoriamente con 1 a 6 atributos binarios.

Palabras claves: *Naive Bayes, regresión logística, clasificación, aprendizaje supervisado.*

Abstract: In data science, most classification models fall into the category of either discriminative models or generative models. Discriminative models only capture the relationship between the attributes of an instance and its class, whereas generative models seek to represent the entire data distribution. Although most classification models are discriminative, it cannot be assured that this type of models is better than generative models. In that sense, the comparison of Naive Bayes and Logistic Regression algorithms as very representative models of discriminative and generative classifiers, respectively, is addressed. In this work, the accuracy of Naive Bayes and Logistic Regression models are evaluated as a function of the number of attributes and instances of an artificial dataset, where both attributes and classes are binary. Unlike other methodologies that employ the datasets to approximate the classification error, this work only employs the datasets to perform the training of the models, while the classification error is computed exactly for the distribution of the data. Experiments show a binary classification accuracy that tends to be slightly better for Logistic Regression using 50 to 500 training instances, when we average the results of randomly generated distributions with 1 to 6 binary attributes.

Key words: *Naive Bayes, logistic regression, classification, supervised learning.*

Introducción

Los clasificadores generativos son modelos que aprenden una probabilidad conjunta $P(X, C)$, definida a partir de las entradas X y la clase C. Estos modelos hacen sus predicciones usando la regla de Bayes para calcular $P(C \mid X)$ y luego seleccionan la etiqueta más probable C. En cambio, los clasificadores discriminativos modelan a $P(C \mid X)$ directamente, o aprenden un mapeado directamente de las entradas X a las etiquetas de clase C. Existe

Recibido: 29/11/2021 Aceptado: 01/02/2022

un mayor número de modelos discriminativos en la literatura que modelos generativos, esto se debe en buena medida a que la obtención de un modelo generativo es una tarea más general que un modelo discriminativo (Ng & Jordan, 2002; Vapnik, 1998). Sin embargo, no existe un consenso de que los modelos discriminativos sean estrictamente mejores que los modelos generativos.

El modelo Naive Bayes es un clasificador generativo muy representativo por la gran cantidad de aplicaciones que posee (Webb *et al*., 2010), siendo un modelo de gran simplicidad bajo el supuesto de independencia condicional entre cada par de atributos dado el valor de la clase. El modelo de Regresión Logística también es muy representativo para el caso de los clasificadores discriminativos, dada su simplicidad y gran cantidad de aplicaciones (Hilbe, 2009). Por lo tanto, la comparación de la Regresión logística con Naive Bayes es un abordaje empleado para analizar las ventajas y desventajas de la elección de modelos generativos o discriminativos en tareas de clasificación.

La comparación de los modelos Naive Bayes y Regresión Logística ha sido abordada tanto de forma directa donde el objetivo es la comparación, como de forma indirecta donde la comparación solo se realiza para identificar los mejores clasificadores para resolver un problema específico. En cuanto a trabajos cuyo objetivo principal es la comparación de modelos, Ng & Jordan (2002) muestran experimentos donde Naive Bayes tiende a tener menor error de acierto con pocos atributos de entrenamiento, pero que es superado por la regresión logística cuando consideramos el comportamiento asintótico del error. Sin embargo, existen otras medidas de evaluación de clasificación como precisión, exhaustividad, error absoluto medio, error absoluto medio promedio y medida kappa, donde la regresión logística tiende a ser superior según los conjuntos de datos evaluados por Gladence *et al*. (2015). En cuanto a comparaciones indirectas de Naive Bayes y Regresión Logística encontramos una mayor cantidad de trabajos, dado que ambos son modelos muy empleados. Se ha identificado que la Regresión Logística tiende a ser superior en las principales métricas de clasificación a Naive Bayes en detección de fraude en tarjetas de crédito (Itoo *et al*., 2021), clasificación de sentimiento (Prabhat & Khullar, 2017), clasificación de texto (Pranckevičius & Marcinkevičius, 2017), identificación de genotipos de maíz (Seka *et al*., 2019), detección de cáncer de mama (Sehgal *et al*., 2012), detección de trastornos de ansiedad (van Eeden *et al*., 2021), identificación de autoría en texto (Aborisade & Anwar, 2018) y discriminación de eventos sísmicos (Dong *et al*., 2016). Mientras que no se ha detectado una diferencia tan significativa entre ambos modelos para problemas de predicción de deslizamientos (Nhu *et al*., 2020), evaluar la necesidad de angiografía coronaria (Golpour *et al*., 2020), detección de spam (Othman & Din, 2019), detección de diabetes (Utami *et al*., 2021), clasificación de préstamos (Pundlik, 2016) y predicción de mortalidad por quemaduras (Stylianou *et al*., 2015). Sin embargo se ha detectado una ventaja significativa de Naive Bayes sobre la Regresión Logística en clasificación de rendimiento académico (Chiok, 2017) y detección de fallos en transformadores (Aborisade & Anwar, 2018).

En este trabajo comparamos el desempeño de los algoritmos Naive Bayes y Regresión logística en problemas de clasificación binaria con atributos binarios, donde analizamos la exactitud de ambos clasificadores en función al número de atributos y tamaño de la muestra. Dado que no es fácil conseguir una cantidad significativa de conjuntos de datos reales cumpliendo las características señaladas, se generaron artificialmente 10 distribuciones probabilísticas para cada $t = 1, 2, \ldots, 6$ atributo binario y a partir de cada distribución se generaron 10 conjuntos de datos de $K = 50, 100, \ldots, 500$ instancias. Los conjuntos de datos fueron empleados solamente para entrenar a los modelos. Las distribuciones no fueron consistentes ya que a cada configuración de atributos les puede tocar distintas clases, por lo tanto a la hora de evaluar la exactitud de los modelos se asume que la categoría correcta para una configuración de atributos es la que aparece con más probabilidad. Mientras que

lo usual es aproximar el error de clasificación a través de un conjunto de testeo o aplicando validación cruzada, este trabajo calcula el error exacto de clasificación empleando la distribución generada y una fórmula que pondera el error de clasificación del modelo en cada configuración posible de atributos. Por lo tanto, el aporte de este trabajo consiste en los siguientes puntos:

- -Una metodología para generar distribuciones artificiales de datos, conjuntos de datos artificiales y el cálculo exacto del error del modelo al ser entrenado en cada conjunto de datos.
- -La aplicación de la metodología para evaluar la exactitud de los modelos Naive Bayes y Regresión logística, además de como la exactitud es afectada por el número de atributos o instancias de entrenamiento.

Este artículo se organiza de la siguiente manera. Primero tenemos una sección donde presentamos las nociones y notaciones empleadas para describir la generación de conjuntos de datos. En la sección siguiente describimos la generación de datos artificiales. Después se detalla la evaluación de modelos y los resultados numéricos. Finalmente se presentan las conclusiones sobre cómo estos resultados se relacionan con lo que ya existe en la literatura.

Materiales y Métodos

Nociones preliminares

Consideremos el problema en el que serán estudiados t atributos, los cuales serán representados mediante X_1, X_2, …, X_t y además, se considera que dichos atributos permiten predecir el valor de la clase C. En este caso, los atributos son considerados como las variables independientes y la clase es la variable dependiente del problema.

Con esto, si consideramos que la variable o atributo X_i posee n_i valores posibles de ocurrencia, donde $1 \leq i \leq t$, y la clase C tiene n_c posibilidades de ocurrencia, entonces el número total de combinaciones será igual a:

$$n = \left(\prod_{i=1}^{t} n_i \right) n_C \quad (1)$$

Cada valor que puede tomar la variable X_i será simbolizado mediante $x_{i;j}$, donde $1 \leq j \leq n_i$. Dados los atributos X_1, X_2, …, X_t y la clase C, un patrón consiste en un elemento del conjunto

$$E = X_1 \times X_2 \times \ldots \times X_t \times C.$$

Mientras que una instancia de un conjunto de datos consiste en un patrón perteneciente a un conjunto de datos que fue generado por una distribución probabilística D en E. Con esto, si tenemos t atributos y una clase, entonces la cantidad total de patrones posibles viene dada por la ecuación (1).

Si a cada patrón $P_i \in E$ le asignamos un número aleatorio N_i, entonces a cada patrón p_i también le podemos asociar una probabilidad P_i de la siguiente manera

$$P_i = \frac{N_i}{\sum_{j=1}^{n} N_j} \quad (2)$$

donde n representa la cantidad de patrones posibles y viene dada por la ecuación (1).

Notemos que hay n vectores o patrones en E, y cada vector o patrón tiene $t + 1$ entradas o variables (entre atributos y clases), entonces una manera de representar todos los vectores o patrones posibles es mediante una representación matricial $n \times (t + 1)$, donde cada fila puede ser asociada unívocamente con un patrón de E.

El conjunto de distribuciones que es utilizado para la prueba de sensibilidad de los algoritmos es conocido como Distribución Original. Mientras que el conjunto compuesto de datos generados de manera automática y aleatoria, que sirve para el entrenamiento de los algoritmos es conocido como Datos Aleatorios.

Métodos para generación de datos

En esta sección describimos la metodología para obtener los datos de prueba que serán empleados

para comparar los modelos Naive Bayes y Regresión Logística. Esta metodología se basa en dos etapas principales i) generar distribuciones sobre atributos y una clase binarios y ii) generar conjuntos de datos a partir de cada distribución obtenida en la etapa previa.

Generación de distribuciones

Para la construcción de la tabla o matriz que representará la Distribución Original seguiremos los siguientes pasos:

Paso 1: Determinar la cantidad de atributos.

Paso 2: Determinar la cantidad de filas y columnas que tendrá la tabla.

Paso 3: Crear la tabla de combinaciones de valores de las variables.

Paso 4: Asignar a cada fila de la tabla de combinaciones posibles un número aleatorio.

Paso 5: Convertir los números aleatorios en probabilidades.

Ejemplo: Consideremos un caso con tres atributos binarios y con una única clase que también sea binaria. Utilizando los siguientes símbolos:

$X1$:= Atributo 1;
$X2$:= Atributo 2;
$X3$:= Atributo 3;
C := Clase.

Este caso lo podemos presentar con el formato de la Tabla 1.

Tabla 1. Esquema de conjunto de datos para tres atributos y una clase binaria.

X_1	X_2	X_3	C
2 opciones	2 opciones	2 opciones	2 opciones

Como consideramos un caso en el que los atributos y la clase son binarios, es decir que en cada una de ellas tenemos dos opciones de ocurrencia, entonces

$$n_1 = n_2 = n_3 = n_C = 2,$$

así, si queremos conocer la cantidad total de combinaciones posibles de valores de estas variables, aplicando la ecuación (1), tendremos

$$n_1 \cdot n_2 \cdot n_3 \cdot n_C = 16,$$

Con esto sabemos que necesitaremos de 16 filas para presentar todas las posibles combinaciones de valores de las variables.

Si consideramos los datos del clima como atributos: la Presión, la Humedad, la Temperatura; y como clase: la Lluvia. Al simbolizar estas variables categóricas tendremos:

$X1$:= Presión;
$X2$:= Humedad;
$X3$:= Temperatura;
C := Lluvia.

Ahora debemos ver cómo varían las categorías de cada atributo y el de la clase. Así podemos tener la Tabla 2.

Tabla 2. Atributos y clases junto a sus posibles valores.

Variables	**Valores**
X_1 = Presión	X11= Alta
	X12 = Baja
X_2 = Humedad	X21 = Mínima
	X22 = Máxima
X_3 = Temperatura	X31 = Calor
	X32 = Frio
Clase	**Valores**
C = Lluvia	C11 = Si
	C12 = No

Al combinar todas las categorías de las variables entre sí, obtenemos la Tabla 3.

Tabla 3. Tabla con todos los patrones posibles siguiendo el ejemplo.

X_1	X_2	X_3	C
Alta	Mínima	Calor	Si
Alta	Mínima	Calor	No
Alta	Mínima	Frío	Si
Alta	Mínima	Frío	No
Alta	Máxima	Calor	Si
Alta	Máxima	Calor	No
Alta	Máxima	Frío	Si
Alta	Máxima	Frío	No
Baja	Mínima	Calor	Si
Baja	Mínima	Calor	No
Baja	Mínima	Frío	Si
Baja	Mínima	Frío	No
Baja	Máxima	Calor	Si
Baja	Máxima	Calor	No
Baja	Máxima	Frío	Si
Baja	Máxima	Frío	No

Ahora asignamos un número aleatorio a cada patrón, con esto obtenemos la Tabla 4.

Tabla 4. Lista de patrones con los números aleatorios asignados.

X_1	X_2	X_3	C	N_i
Alta	Mínima	Calor	Si	0.02
Alta	Mínima	Calor	No	0.04
Alta	Mínima	Frío	Si	0.3
Alta	Mínima	Frío	No	0.07
Alta	Máxima	Calor	Si	0.1
Alta	Máxima	Calor	No	0.08
Alta	Máxima	Frío	Si	0.21
Alta	Máxima	Frío	No	0.13
Baja	Mínima	Calor	Si	0.4
Baja	Mínima	Calor	No	0.05
Baja	Mínima	Frío	Si	0.24
Baja	Mínima	Frío	No	0.38
Baja	Máxima	Calor	Si	025
Baja	Máxima	Calor	No	0.58
Baja	Máxima	Frío	Si	0.7
Baja	Máxima	Frío	No	0.09

Por último, convertimos los números aleatorios en probabilidades, utilizando la ecuación (2), agregamos el número *i* indicador de fila, obteniendo así la Tabla 5.

Tabla 5. Distribución finalmente generada por la metodología.

i	X_1	X_2	X_3	C	P_i
1	Al.	Mín.	Calor	Si	0.00549451
2	Al.	Mín.	Calor	No	0.01098901
3	Al.	Mín.	Frío	Si	0.08241758
4	Al.	Mín.	Frío	No	0.01923077
5	Al.	Máx.	Calor	Si	0.02747253
6	Al.	Máx.	Calor	No	0.02197802
7	Al.	Máx.	Frío	Si	0.05769231
8	Al.	Máx.	Frío	No	0.03571459
9	Ba.	Mín.	Calor	Si	0.10989011
10	Ba.	Mín.	Calor	No	0.01373626
11	Ba.	Mín.	Frío	Si	0.06593407
12	Ba.	Mín.	Frío	No	0.10439560
13	Ba.	Máx.	Calor	Si	0.06868132
14	Ba.	Máx.	Calor	No	0.15934066
15	Ba.	Máx.	Frío	Si	0.19230769
16	Ba.	Máx.	Frío	No	0.02472527

Generación de conjuntos de datos

Para la construcción de la tabla con los datos aleatorios necesitaremos definir algunos parámetros. Para decidir si el *i*-ésimo patrón formará parte de los datos de prueba, tendremos en cuenta la probabilidad P_i asignada a dicho patrón en la Distribución Original, y la misma será utilizada para definir el rango de valores, el cual estará determinado por un límite inferior LI_i y un límite superior LS_i. Una vez que tengamos éstos límites podemos utilizar un número aleatorio *a* producido al azar, éste número *a* lo compararemos con los límites inferiores y superiores, y al rango al que pertenezca este número aleatorio será el patrón seleccionado para formar parte de los datos de prueba. Generamos *a* con una

distribución uniforme en un rango entre 0 y 1.

Considerando las siguientes notaciones:

i := posición del patron en la Distribución Original;

Ri := rango i;

LIi := límite inferior del rango i;

LSi := límite superior del rango i;

Pi := probabilidad del patron i;

$AcPi$:= vector acumulador de probabilidad hasta el patron i.

Podemos definir el vector acumulador de probabilidad como:

$$\mathrm{AcP_i} = \begin{cases} \mathrm{P_1} & \text{si} \quad \mathrm{i = 1;} \\ \mathrm{AcP_{i-1} + P_i} & \text{si} \quad \mathrm{i > 1.} \end{cases} \quad (3)$$

Para hallar los límites inferior y superior, consideraremos los siguientes casos:

- Para la fila $i = 1$, consideramos

$$\begin{cases} \mathrm{LI_1} = 0; \\ \mathrm{LS_1} = \mathrm{AcP_1}. \end{cases} \quad (4)$$

- Para $2 \leq i \leq n - 1$, consideramos

$$\begin{cases} \mathrm{LI_i = AcP_{i-1}} \quad \text{y} \quad \mathrm{LI_i = LS_{i-1}}; \\ \mathrm{LS_i} = \mathrm{AcP_i}. \end{cases} \quad (5)$$

- Para $i = n$, consideramos

$$\begin{cases} \mathrm{LI_n = AcP_{n-1}} \quad \text{y} \quad \mathrm{LI_n = LS_{n-1}}; \\ \mathrm{LS_n} = 1. \end{cases} \quad (6)$$

Mientras que el rango del patrón i estará definido como:

$$Ri = (LI_i, LS_i]. \quad (7)$$

Es importante tener en cuenta que se ha agregado el vector acumulador de probabilidad para definir los límites inferior y superior de cada rango y entre estos límites estará ubicado el número aleatorio a que se utilizará para seleccionar la instancia. Así, si a es el número aleatorio generado por el programa, entonces buscamos los límites inferior y superior tal que

$$LI_i < a \leq LS_i. \quad (8)$$

La cantidad de filas de la tabla con los datos aleatorios será igual a la cantidad de elementos de la muestra indicada, esto quiere decir que si queremos una muestra con K instancias, entonces la tabla con los datos aleatorios tendrá K filas. Mientras que la cantidad de columnas coincidirá con la cantidad de columnas de la Tabla correspondiente a la Distribución Original.

El tamaño de la tabla con los datos aleatorios estará determinado por la cantidad de filas o instancias, ya que la cantidad de columnas será la misma en ambas tablas (en la tabla de Distribución Original y en la tabla con los Datos Aleatorios).

Por último, la Distribución Aleatoria será construida siguiendo los siguientes pasos:

Paso 1: Generar un número aleatorio a_j (la j indica que es el j-ésimo valor generado).

Paso 2: Verificar en que rango de valores se ubica el número generado:

$$LI_i < a \leq LS_i,$$

donde el valor de i nos indicará la fila en la tabla de la Distribución Original (el patrón) a ser seleccionada.

Ejemplo: Siguiendo con el Ejemplo anterior, lo primero que debemos hacer es presentar la Tabla de la Distribución Original de tal manera que se puedan observar las probabilidades acumuladas. Esta idea es presentada en la Tabla 6.

Supongamos que ya generamos varios valores a_j,

- Si $a_1 = 0.54$, entonces buscamos el valor de i que verifique

$$LI_i < a_1 \leq LS_i,$$

con lo que

$$LI_i < 0.54 \leq LS_i,$$

lo cual implica que $i = 11$, ya que

$$0.54 < 0.55494505 = AcP_{11}.$$

Tabla 6. Lista de patrones con sus intervalos de selección.

Ditribución Original						AcP_i
i	X_1	X_2	X_3	C	P_i	
1	Al.	Mí.	Calor	Si	0.00549451	0.00549451
2	Al.	Mí.	Calor	No	0.01098901	0.01648352
3	Al.	Mí.	Frío	Si	0.08241758	0.09890110
4	Al.	Mí.	Frío	No	0.01923077	0.11813187
5	Al.	Má.	Calor	Si	0.02747253	0.14560440
6	Al.	Má.	Calor	No	0.02197802	0.16758242
7	Al.	Má.	Frío	Si	0.05769231	0.22527473
8	Al.	Má.	Frío	No	0.03571459	0.26098901
9	Ba.	Mí.	Calor	Si	0.10989011	0.37087912
10	Ba.	Mí.	Calor	No	0.01373626	0.38461538
11	Ba.	Mí.	Frío	Si	0.06593407	0.45054945
12	Ba.	Mí.	Frío	No	0.10439560	0.55494505
13	Ba.	Má.	Calor	Si	0.06868132	0.62362637
14	Ba.	Má.	Calor	No	0.15934066	0.78296703
15	Ba.	Má.	Frío	Si	0.19230769	0.97527473
16	Ba.	Má.	Frío	No	0.02472527	1

Con esto, el patrón seleccionado es el presentado en la Tabla 7.

Tabla 7. Primer patrón seleccionado.

i	X_1	X_2	X_3	C
1	Baja	Mínima	Frío	Si

- Si $a_1 = 0.14$, entonces $i = 11$, esto se obtiene siguiendo el procedimiento indicado anteriormente. Con esto, el patrón seleccionado es el presentado en la Tabla 8.

Tabla 8. Segundo patrón seleccionado.

i	X_1	X_2	X_3	C
4	Alta	Mínima	Frío	No

Con estos dos patrones seleccionados formamos la Tabla 9, en la misma se presentan los datos aleatorios.

Tabla 9. Tabla generada con los dos primeros patrones.

X_1	X_2	X_3	C
Baja	Mínima	Frío	Si
Alta	Mínima	Frío	No

Este proceso se debe repetir hasta tener K filas para la tabla de datos aleatorios, la cual se tomará como la muestra de entrenamiento para los algoritmos.

Resultados y discusión

Los algoritmos fueron evaluados en conjuntos de datos de m = 1,2,3,4,5,6 atributos. Para cada m se generaron 10 distribuciones aleatorias y para cada

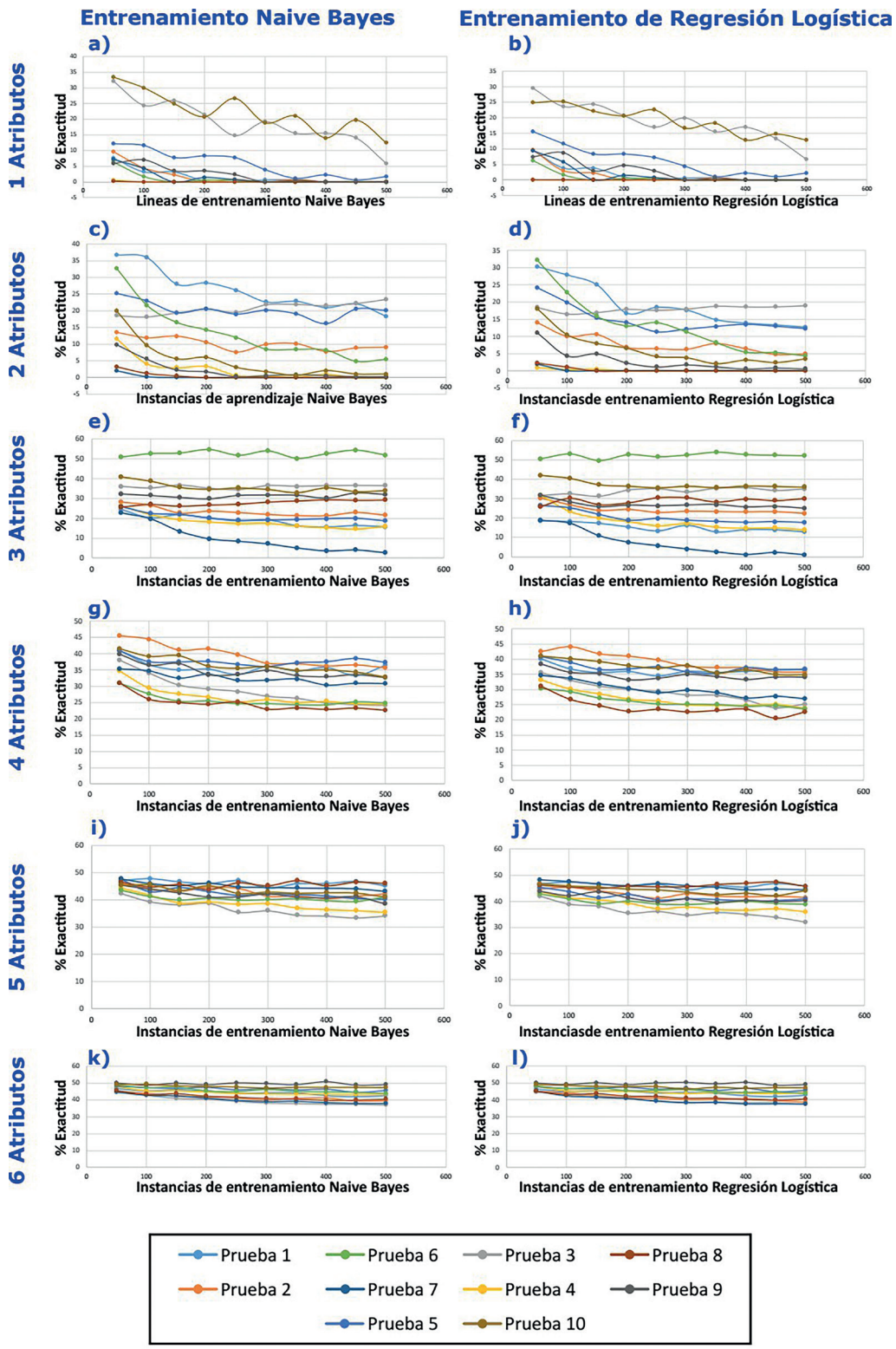

Figura 1: Comparación de exactitud de los algoritmos en las distribuciones por número de atributos.

distribución aleatoria se generaron 10 conjuntos de datos de $K = 50, 100, \ldots, 500$ instancias. En vez de emplear conjuntos de testeo o validación cruzada para estimar el error, este trabajo calcula el error real de los modelos aprovechando que se conoce la distribución real de los datos. Puede notarse también que las distribuciones tienden a producir distintas clases para una misma asignación de valores de los atributos, por lo que si esperamos que el modelo prediga la clase a partir de los atributos siempre existirá un error. Por lo tanto, el cálculo del error de cada modelo se realiza de forma exacta asumiendo que a) la correcta del modelo para cada configuración de atributos es la moda de la clase para tal configuración y b) el tamaño del conjunto de prueba tiende a infinito. Eso se consigue a partir de la siguiente formula

$$\varepsilon = \sum_{e \in E} \delta\left(M\left(e\right), m_e\right) P_e \,, \quad (9)$$

donde i) $M(e)$ es la salida del modelo de clasificación tomando los valores de los atributos del patrón e de entrada, ii) m_e es la moda de la clase C para la configuración de atributos del patrón e, iii) P_e es la probabilidad de ocurrencia del patrón e y iv) $\delta(x, y)$ es una función igual a 1 si $x = y$ e igual a 0 caso contrario. Por lo tanto la exactitud de la clasificación se puede calcular como $1 - \varepsilon$ a partir de la ecuación (9).

La Figura 1 sumariza los resultados experimentales, donde cada subfigura presenta la exactitud de cada algoritmo aplicado sobre las distribuciones correspondiente a cada número de atributos específico, donde el número atributos crece a medida que bajamos y cada columna de sub-gráficos corresponde a un algoritmo. Los sub-gráficos correspondientes a un mismo número de atributos mantienen el color correspondiente a cada distribución generada, de modo a que se puedan comparar las curvas tanto para Naive Bayes como para la Regresión Logística. Para ambos algoritmos puede notarse como las curvas tienden a descender y acercarse entre sí a medida que crece el número de atributos. También puede notarse como ambos algoritmos se comportan de manera similar para las mismas distribuciones, lo que evidencia que el error no varía proporcionalmente demasiado de un algoritmo a otro. Por lo tanto si una distribución produce más error que otra para un algoritmo, tiende a ocurrir lo mismo en el otro algoritmo, según nuestros experimentos. La Figura 2 presenta la exactitud para cada algoritmo promediando entre las distribuciones generadas para cada número de atributos. Se siguen observando patrones muy

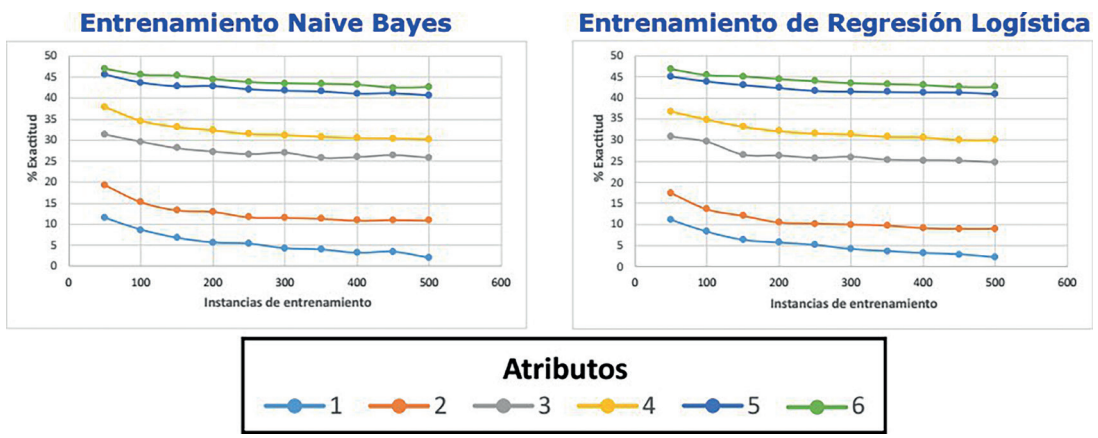

Figura 2: Comparación de la exactitud promedio de los algoritmos.

similares para ambos algoritmos.

La Tabla 10 presenta la diferencia entre Regresión Logística y Naive Bayes del promedio de exactitud entre las distribuciones de cada número fijo de atributos, en función al número de instancias de entrenamiento. Puede notarse que la diferencia

Tabla 10. Diferencias de porcentajes de error promedio de predicción de los algoritmos.

Nro. de atributos	Instancias de entrenamiento									
	50	100	150	200	250	300	350	400	450	500
1	0.26	0.31	0.36	-0.01	0.18	0.08	0.27	-0.04	0.50	-0.18
2	1.92	1.74	1.32	2.55	1.49	1.56	1.62	1.76	2.04	1.92
3	0.48	0.04	1.59	1.00	0.94	1.00	0.49	0.79	1.37	1.19
4	1.10	-0.26	-0.07	0.24	-0.11	-0.09	0.03	-0.14	0.44	0.19
5	0.48	-0.24	-0.27	0.45	0.39	0.19	0.15	-0.28	-0.18	-0.30
6	0.16	0.15	0.20	0.03	-0.18	-0.01	0.07	0.07	-0.12	0.02

nunca supera 1 y que la mayoría de las veces el valor es positivo, lo que implica que la mayoría de las veces Naive Bayes tuvo un error mayor a la Regresión Logística pero sin superar un 1 %.

Es importante mencionar que la Prueba i no es la misma para cada fila, son pruebas con distintas distribuciones pero numeradas de forma similar por lo que no guardan relación.

Conclusión

Los resultados numéricos no implican que la Regresión Logística sea siempre superior a Naive Bayes. Sin embargo, se ve una leve tendencia a favor de la Regresión Logística y los experimentos donde Naive Bayes supera a la Regresión Logística representan el caso menos común. Estas observaciones siguen la tendencia de los trabajos consultados en la literatura, donde se comparan ambos modelos para solucionar problemas reales. Sin embargo, los experimentos realizados no muestran que el número de instancias revierta de alguna forma la tendencia observada. Esto no se condice con el trabajo de Ng y Jordan , donde se argumenta que el error en Naive Bayes converge más rápidamente a su valor asintótico en función del tamaño de la muestra, aunque la Regresión Logística tendría valores asintóticos menores para el error.

Una ventaja de la Regresión Logística al modelo Naive Bayes representa evidencia importante a favor de los modelos discriminantes, aunque tal diferencia no sea de gran magnitud. Esto se debe a que la Regresión Logística es solo un modelo lineal y por lo tanto está entre los de menor capacidad. Mientras que Naive Bayes es un modelo muy empleado entre los modelos generativos por la generalidad de sus hipótesis.

Posibles extensiones de este trabajo son:

- Extender las pruebas con distribuciones cuyos atributos no sean solo binarios, clasificación multi-clase, mayor número de atributos o distribuciones satisfaciendo hipótesis particulares.
- Comparación con modelos generativos más sofisticados que Naive Bayes.

Contribución de los autores

Los autores contribuyeron de igual manera en la elaboración de este artículo.

Conflictos de interés

Los autores declaran no tener conflictos de interés.

Bibliografía

Aborisade, O. & Anwar, M. (2018). Classification for authorship of tweets by comparing logistic regression and naive bayes classifiers. Pp. 269–276, *in* Bilof, R. (Ed.). *Proceedings of the 2018 IEEE 19th International Conference on Information Reuse and Integration for Data Science*. Salt Lage City: IEEE Computer Society Conference Publishing Services. 546 pp.

Chiok, C. (2017). Predicción del rendimiento académico aplicando técnicas de minería de datos. *Anales Científicos*, 78(1): 26–33.

Dong, L., Wesseloo, J., Potvin, Y. & Li, X. (2016).

Discrimination of mine seismic events and blasts using the fisher classifier, naive bayesian classifier and logistic regression. *Rock Mechanics and Rock Engineering,* 49(1): 183–211.

Gladence, L., Karthi, M. & Anu, V. (2015). A statistical comparison of logistic regression and different Bayes classification methods for machine learning. *ARPN Journal of Engineering and Applied Sciences,* 10(14): 5947–5953.

Golpour, P., Ghayour-Mobarhan, M., Saki, A., Esmaily, H., Taghipour, A., Tajfard, M., Ghazizadeh, H., Moohebati, M. & Ferns, G. (2020). Comparison of Support Vector Machine, Naïve Bayes and Logistic Regression for Assessing the Necessity for Coronary Angiography. *International Journal of Environmental Research and Public Health,* 7(18)6449: 1–9.

Hilbe, J.M. (2009). *Logistic regression models.* London: Chapman and hall/CRC. xviii + 638 pp.

Itoo, F., Meenaksi & Singh, S. (2021). Comparison and analysis of logistic regression, Naïve Bayes and KNN machine learning algorithms for credit card fraud detection. *International Journal of Information Technology,* 13(4): 1503–1511.

Ng, A. & Jordan, M. (2002). On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes. Pp. 841–848, *in* Dietterich, T.G., Becker, S. & Ghahramani, Z. (Eds.). *Proceedings of the 14th International Conference on Neural Information Processing Systems: Natural and Synthetic*. Cambridge: MIT Press. 1594 pp.

Nhu, V.-H., Shirzadi, A., Shahabi, H., Singh, S., Al-Ansari, N., Clague, J., Jaafari, A., Chen, W., Miraki, S., Dou, J., Luu, C., Górski, K., Pham, B.T., Nguyen, H.D. and Ahmad, B.B. (2020). Shallow landslide susceptibility mapping: A comparison between logistic model tree, logistic regression, naïve bayes tree, artificial neural network, and support vector machine algorithms. *International journal of environmental research and public health*, 17(8)2749: 1–30.

Othman, N. & Din, W. (2019). Youtube spam detection framework using naive bayes and logistic regression. *Indonesian Journal of Electrical Engineering and Computer Science*: 14(3), 1508–1517.

Prabhat, A. & Khullar, V. (2017). Sentiment classification on big data using Naïve Bayes and logistic regression. Art. CS364, pp 1–5 *in* Sri Shakthi Institute of Electrical and Electronics Engineers (Ed.). *Proceedings of the 2017 International Conference on Computer Communication and Informatics (ICCCI)*. Coimbatore: Sri Shakthi Institute of Electrical and Electronics Engineers. 665 pp.

Pranckevičius, T. & Marcinkevičius, V. (2017). Comparison of naive bayes, random forest, decision tree, support vector machines, and logistic regression classifiers for text reviews classification. *Baltic Journal of Modern Computing*, 5(2): 221–232.

Pundlik, R. (2016). Comparison of sensitivity for consumer loan data using Gaussian Naïve Bayes (GNB) and logistic regression (LR). Pp. 120–124, *in* Al-Dabass, D., Achalakul, T., Sarochawikasit, R. & Prom-On, S. (Eds.). *Proceedings of the 2016 7th International Conference on Intelligent Systems, Modelling and Simulation (ISMS)*. Salt Lage City: IEEE Computer Society Conference Publishing Services. xvii + 456 pp.

Sehgal, C., Cary, T., Cwanger, A., Levenback, B. & Venkatesh, S. (2012). Combined Naïve Bayes and logistic regression for quantitative breast sonography. Pp. 1686-1689, *in* Weihnacht, M. & Schmidt, H. (Eds.). *2012 IEEE International Ultrasonics Symposium*. Dresden: Institute of Electrical and Electronics Engineers. 2772 pp.

Seka, D., Bonny, B., Yoboué, A., Sié, S. & Adopo-Gouréne, B. (2019). Identification of maize

(*Zea mays* L.) progeny genotypes based on two probabilistic approaches: Logistic regression and naïve Bayes. *Artificial Intelligence in Agriculture*, 1: 9–13.

Stylianou, N., Akbarov, A., Kontopantelis, E., Buchan, I. & Dunn, K. (2015). Mortality risk prediction in burn injury: Comparison of logistic regression with machine learning approaches. *Burns*, 41(5): 925–934.

Utami, D., Nurlelah, E. & Hasan, F. (2021). Comparison of Neural Network Algorithms, Naive Bayes and Logistic Regression to predict diabetes. *Journal of Informatics and Telecommunication Engineering*, 5(1): 53–64.

Van Eeden, W., Luo, C., van Hemert, A., Carlier, I., Penninx, B., Wardenaar, K., Hoos, H. & Giltay, E. (2021). Predicting the 9-year course of mood and anxiety disorders with automated machine learning: A comparison between auto-sklearn, naïve Bayes classifier, and traditional logistic regression. *Psychiatry Research*, 299(113823): 1–10.

Vapnik, V. (1998). *Statistical learning theory*. New York: Wiley-Interscience. 768 pp.

Webb, G., Keogh, E. & Miikkulainen, R. (2010). Naïve Bayes. Pp. 713–714, in Sammut, C. & Webb, G.I. (Eds.). *Encyclopedia of Machine Learning*. New York: Springer Science & Business Media. 1031 pp.