Análisis del rendimiento del algoritmo de clasificación Naive Bayes vs C4.5 (Árbol de decisión)

José Massón[1]

***Abstract*— La clasificación supervisada es muy utilizada para etiquetar registros que más adelante serán utilizados como entrenamiento para considerar nuevos registros a ser clasificados. Los algoritmos de clasificación basados en árboles son muy buenos para realizar predicciones. Sin embargo, su efectividad depende del tipo de atributo y su respectivo valor que tenga cada columna del dataset utilizado. Se buscará predecir la efectividad del Naive Bayes y J48 dependiendo de los datasets que se le pasen como entrada a partir de datasets con los cuales se dio el entrenamiento.**

I. Introducción

Hay que considar que la extracción de los datos no es específica para un tipo de dato. La minería de datos debe ser aplicable a cualquier tipo de repositorio de información. Ésta se está usando con mayor frecuencia, y se la aplica más en bases de datos, sean estas *relacionales*, *objeto-relacionales*, entre otras.

La minería de datos tiene diversas funciones que se encuentran clasificadas como: clasificación, agrupamiento *(clustering)*, seleccin de características, etcétera.

Sin embargo, en este artículo se centra en la clasificación de los datos y la medición del rendimiento de algunos algoritmos clasificadores basados en la tasa de TP *(True Positive)*, FP *(False Positive)*, la exactitud *(Accuracy)*. Éstas métricas son generadas por los algoritmos cuando son aplicados al conjunto de datos que se quieren *clasificar.*

La clasificación es la organización de los datos en clases. Se la conoce también como *clasificación supervisada*, ya que utiliza etiquetas para ordenar los objetos en la recopilación de los datos. Utiliza un conjunto de entrenamiento donde todos los objetos ya están asociados con etiquetas de clase. El algoritmo de clasificación *aprende* de este conjunto de entrenamiento y construye el modelo para predecir la etiqueta de clase de *nuevos objetos*.

II. Trabajos Relacionados

Se han realizado investigaciones referentes a clasificación en algunos aspectos. Se los utiliza frecuentemente para realizar predicciones, y ahora que la era del Internet, la digitalización y almacenamiento masivo de los datos en la nube, junto con el gran poder que actualmente tienen los ordenadores, permiten que se realizan investigaciones y análisis en un menor tiempo.

Se debe tener claro que para *Naive Bayes*, se debe conocer acerca del *Teorema de Bayes* [1] [2], mismo que permitir entender lo que realiza el algoritmo.

Algunas aplicaciones de este clasificador se aprecian en la medicina [3]. En esta investigación, se analizaron 3 datasets que contenían información acerca de si un paciente con cáncer al pulmón era sobreviviente a partir de datos como el estado del cáncer **(benigno o maligno)**, el tiempo de la enfermedad presente en el paciente, la fase en la que se encontraba el cáncer, entre otros. Cabe destacar que estos atributos son discretos y categóricos, sobre todo el de la clase, el cual es binario, y de ahí la naturaleza por tratarlo como árbol binario, facilitando muchas simplificaciones al momento de realizar una predicción.

Aquí se trabajó con el algoritmo de Bayes, y también como un árbol de decisión J48. Cuando se lo trabajó de esta forma, se detallaron las reglas [4] [3] que consideraban la clasificación de un registro, a tal punto que si no es válido, era eliminado del conjutno de instancias que se consideraba limpia para ser procesado.

Se realizaron las comparaciones respectivas a través de la exactitud con la cual clasificacó los datos, ya que lo que se busca es la menor cantidad de falsos positivos que arroje como resultado cada algoritmo. Además, con la mayor exactitud posible y la menor cantidad de falsos positivos se puede determinar cuál de los dos es el mejor algoritmo de clasificación. Sin embargo, se tienen que considerar los atributos y los valores de los mismos para cada dataset, ya que por ejemplo aquí, no hubieron valores continuos, sino mas bien discretos, identificando de manera concisa que información se almacenaba en cada registro.

III. Recolección de los datos

Los datasets considerados para esta investigacin son 8 aproximadamente. Ésto se debe a que se desea trabajar con atributos de diferentes tipos, tales como nominales, categricos, discretos, continuos, para determinar qu algoritmo de clasificacin es mejor para cada tipo de atributo.

Entre los más importantes se encuentran los siguientes:

A. Iris

Cuenta con 151 registros, cuyos atributos son:

Id: Representa el identificador de la instancia.

SepalLengthCm: Representa la longitud del sépalo en centímetros.

SepalWidthCm: Representa la anchura del sépalo en centímetros.

PetalLengthCm: Representa la longitud del pétalo en centímetros.

PetalWidthCm: Representa la anchura del pétalo en centímetros.

Species: Representa la especie a la que pertenece la planta de iris. (Característica que se la considera como clase en este dataset)

B. Mushrooms

Cuenta con 8124 registros, cuyos atributos más importantes son:

Population: Representa cómo se encuentra el hongo en su hábitat natural.

Habitat: Representa el lugar donde se encuentra el hongo.

Spore: Representa el color de la espora del hongo.

Cap-shape: Representa la forma de la parte superior del hongo.

Class: Representa la especie a la que pertenece el hongo. (Característica que se la considera como clase en este dataset). Sus valores son *Comestible - e* y *Venenosa - p*

C. Tic-Tac-Toe Endgame

Place representa los posibles lugares donde se puede encontrar la jugada de acuerdo a su direccin. En este caso sus valores son: *left* - izquierda, *middle* - en medio y *right* - derecha.

A partir de esto, para cada lugar *top, middle y bottom*, se tienen 3 registros, por lo que se tiene un total de 9 atributos, sin incluir la clase.

Considere los posibles valores de las casillas como *x, b* ó *o*. Cuenta con 958 registros, cuyos atributos son:

Top-place-square: Representa la jugada del jugador en la celdas superiores.

Middle-place-square: Representa la jugada del jugador en la celdas intermedias.

Bottom-place-square: Representa la jugada del jugador en la celdas inferiores.

Class: Representa si el jugador que juega con *x* ha ganado la partida. (Característica que se la considera como clase en este dataset). Sus valores son **Positive - ganancia** y **Negative - pérdida**.

IV. Metodología

El dataset descrito anteriormente va a pasar por los dos algoritmos de clasificación: *árboles de decisión y Naive Bayes*.

A. Árboles de decisión

Los árboles de decisión son un método efectivo de aprendizaje supervisado. Su objetivo es la partición de un dataset en grupos homogéneos como sea posible en términos de la variable *(clase)* que va a ser predicha. Este recibe como entrada un conjunto de datos clasificados y arroja un árbol como salida. Éste es un diagrama donde cada nodo *(hoja)* es una decisin y aquellos que son nodos *no finales* representan una prueba. Cada hoja representa la decisión que fue tomada para clasificar el registro despus de haber verificado todas las pruebas presentes desde el nodo raz hasta el nodo hoja. [2]

Los algoritmos populares de árboles de decisión son ID3, C4.5, CART. El algoritmo ID3 se lo considera muy simple. Éste utiliza la ganancia de la información *(information gain)* como criterio de partición.

El algoritmo C4.5 es una evolución del ID3. Utiliza la ganancia de relación *ratio gain* como criterio de partición. Este es el clasificador J48 en Weka.

B. Naive Bayes

El algoritmo de Naive Bayes es un clasificador de probabilidad simple que calcula un conjunto de probabilidades a travs del conteo de la frecuencia y combinaciones de los valores en un dataset dado. Utiliza el Teorema de Bayes y asume que todos los atributos son independientes, dado el valor de la variable de clase. Esta suposicin de independencia condicional raramente es vlida en aplicaciones del mundo real, de ahí toma el nombre de *naive* (ingenuo). Sin embargo, el algoritmo tiende a funcionar bien y aprende rápidamente en problemas de clasificación supervisada. [3]

Los clasificadores de Bayes, a diferencia de las redes neuronales, no tienen varios parmetros libres que se deben establecer. Esto simplifica enormemente el proceso de diseo. Dado que el clasificador devuelve probabilidades, es ms sencillo aplicar estos resultados a una amplia variedad de tareas en vez de usar una escala arbitraria.

Además, no requiere grandes cantidades de datos, por lo que el aprendizaje puede comenzar sin problemas. Hay que considerar que este clasificador es rápido al momento de tomar las decisiones [4] [5].

C. Matriz de confusión

La matriz de confusión contiene información sobre las clasificaciones reales (proporcionado por el dataset) y las predichas, mismas que son hechas por un sistema de clasificación. Para evaluar el rendimiento de estos sistemas, se utilizan los datos presentes en esta matriz.

Se han definido varios términos estándar para esta matriz:
1. **Verdadero positivo (TP):** Si el resultado de una prediccin es p y el valor real también es p, entonces se llama verdadero positivo (TP).

2. **Falso positivo (FP):** Sin embargo, si el valor real es n, entonces se dice que es un falso positivo (FP)

3. **Precisión y recall (TPR):** La precisión es la fraccin de instancias recuperadas que son relevantes, mientras que el recall es la fraccin de instancias relevantes que se recuperan. Tanto la precisin como el recall se basan en una comprensin y medida de la relevancia.

4. La precisión se puede ver como una medida de exactitud o calidad, mientras que la tasa de verdaderos positivos (TPR - True Positive Rate) es una medida de integridad o cantidad.

Un TPR alto significa que un algoritmo devolvió la mayora de los resultados relevantes. En cambio, la alta precisin significa que un algoritmo arroj resultados ms relevantes que irrelevantes.

5. **Exactitud (Accuracy):** Es la métrica definida como la proporción entre las instancias correctamente clasificadas y todas las instancias presentes en el dataset. Esta es una de las métricas que se utilizará para contrastar el rendimiento de clasificación entre los dos algoritmos descritos anteriormente.

V. Resultados

Para cada algoritmo, se consideró trabajar en dos partes. Se dividió cada dataset en dos partes: El 70% de los datos fueron tomados para entrenar al algoritmo de clasificación y la cantidad restante, para la prueba del clasificador.

Gracias a que se trabajó con un porcentaje en la parte de entrenamiento, el algoritmo *conoce más* acerca de los casos que se le pueden presentar, y por ende, aumentar la exactitud en la clasificación de los elementos en la prueba del algoritmo ya entrenado.

Una vez probado los datasets, se obtuvieron los resultados que se muestran en la siguiente tabla:

TABLE I: Comparaciones de Algoritmos de Clasificación

Comparaciones de Exactitudes para cada algoritmo				
Dataset	**Accuracy (C4.5)**		**Accuracy (Naive Bayes)**	
	Train	**Test**	**Train**	**Test**
Bank Marketing	91.10	76.35	80.30	72.30
Credit Card	99.98	100	97.89	97.26
Crime	99.95	93.07	94.57	77.23
Glass	93.33	67.19	55.33	48.44
Iris	100	100	99.05	97.78
Mushrooms	100	100	98.07	94.01
Tic Tac Toe End Game	92.99	80.49	73.03	67.60
Zoo	98.59	76.67	100	86.67

Como se puede apreciar en la tabla, la mayoría de los datasets prefieren utilizar el algoritmo C4.5. Considérese también que solo se está realizando la comparación entre dos algoritmos, y con los datasets más relevantes para el estudio.

Cabe destacar que también se realizaron pruebas con otros datasets que no fueron mencionados en este estudio. Uno de ellos representaba a las transacciones realizadas por los usuarios, y la clase representaba a si iba a realizar un depósito después de haber realizado un número determinado de transacciones y haber excedido un valor tope.

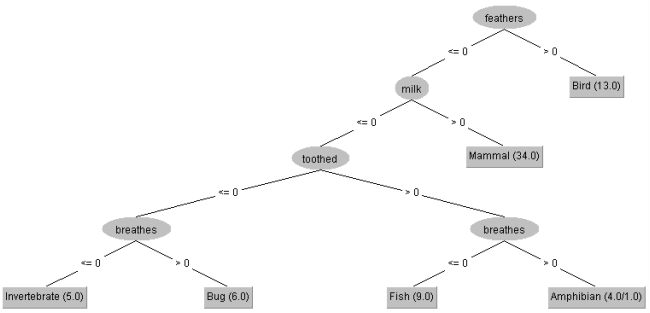

Fig. 1: Árbol de clasificación generado por el algoritmo C4.5 del dataset *Zoo*

En este caso en particular, el algoritmo C4.5 se demoró en procesar el modelo con los datos de entrenamiento, pero la exactitud fue mayor que utilizando el algoritmo de Naive Bayes, sin embargo, éste no difirió mucho, aproximadamente alrededor del 2%.

Con respecto a los datasets en el estudio, se puede apreciar que aquel que trataba sobre los tipos de vidrios, hubo una diferencia radical en cuanto a exactitudes al momento de clasificar los datos. Ésto pudo haber ocurrido por la aleatoriedad de los datos (se trató lo mejor posible que los datos estén distribuidos de manera uniforme en ambos datasets). Cabe destacar que en este, al igual que el mercadeo *Bank Marketing*, los atributos fueron continuos, de ahí que su exactitud sea más bajo en comparación con los demás que fueron testeados.

Se tiene que tener en consideración que se utilizaron dos datasets como ideales, en este caso son *Iris* y *Mushrooms*, que tienen una pequeña diferencia en exactitud entre los métodos utilizados.

El primero contiene atributos continuos, que representan las medidas de algunas estructuras de las flores *iris*, y el segundo, contiene atributos nominales que representan las características de los hongos, tales como su hábitat, color, etcétera.

Se intentó trabajar con datasets que contenían como clase, atributos que sean del tipo discreto o continuo, pero como tal, Naive Bayes y C4.5 soportan sólo clases que sean del tipo nominal y sobre todo, atributos categóricos.

VI. Conclusiones

A partir del estudio realizado con los diferentes algoritmos, y datasets con atributos de tipos variados, se pueden llegar a las siguientes conclusiones:

La más importante de todas, es acerca de los tipos de atributos que tienen los datasets. En el caso del dataset *Tic Tac Toe End Game*, donde todos sus atributos son del tipo nominal, se apreció que la exactitud en clasificar a las instancias superior en el algoritmo C4.5 que utilizando Naive Bayes, lo cual indica que Naive Bayes no es muy bueno al momento de trabajar con datasets donde todos sus atributos sean nominales. Éste mismo caso, se da con el dataset de *Crimes*, donde la mayoría de atributos son del tipo nominal, sin embargo, las diferencias en exactitudes no son tan drásticas como en el caso mencionado anteriormente.

Sin embargo, cuando se trabajan con atributos discretos y continuos, junto con atributos cualitativos nominales, ambos presentes en un solo dataset, los algoritmos no tienen una diferencia en exactitud que dependerá de la cantidad de un tipo de atributos con respecto al otro. En el dataset de *Bank Marketing*, la mayoría de los atributos son *cualitativos nominales* que *cuantitativos*, por lo que el mejor algoritmo para clasificar los registros es el **C4.5**.

Sin embargo, gracias al dataset de *Credit Card*, donde la mayoría de sus atributos son del tipo cuantitativo (ya que son el valor de las transacciones realizados en dos días). Aquí no se apreció mucha diferencia en exactitudes, por lo que es mejor elegir a **Naive Bayes** como algoritmo por excelencia cuando se están trabajando con datos cuantitativos. Considerar que la clase sigue siendo un valor cualitativo y nominal. Éste caso se ve reflejado también en el dataset de **Zoo**, donde los valores de los atributos son cantidades discretas, y la exactitud de Naive Bayes es superior que C4.5

REFERENCES

[1] J. COE, "Performance comparison of naïve bayes and j48 classification algorithms," *International Journal of Applied Engineering Research*, vol. 7, no. 11, p. 2012, 2012.

[2] T. R. Patil and S. Sherekar, "Performance analysis of naive bayes and j48 classification algorithm for data classification," *International Journal of Computer Science and Applications*, vol. 6, no. 2, pp. 256–261, 2013.

[3] G. Dimitoglou, J. A. Adams, and C. M. Jim, "Comparison of the c4. 5 and a naïve bayes classifier for the prediction of lung cancer survivability," *arXiv preprint arXiv:1206.1121*, 2012.

[4] A. Ashari, I. Paryudi, and A. M. Tjoa, "Performance comparison between naïve bayes, decision tree and k-nearest neighbor in searching alternative design in an energy simulation tool," *Int. J. Adv. Comput. Sci. Appl*, vol. 4, no. 11, pp. 33–39, 2013.

[5] J. Han, J. Pei, and M. Kamber, *Data mining: concepts and techniques*. Elsevier, 2011.