Aplicación de algoritmos de clasificación supervisada usando Weka.

Ing. Corso, Cynthia Lorena
Universidad Tecnológica Nacional, Facultad Regional Córdoba

Abstract

Este trabajo está enmarcado en el tema de minería de datos y tiene como finalidad descubrir el nivel de confianza de los algoritmos de clasificación.
Esto se realizará mediante corridas de los algoritmos supervisados de clasificación mostrando el nivel de bondad que tiene los modelos generados.
El interrogante que este trabajo pretende develar es si este nivel de confianza aumenta o disminuye al disponer de mayor cantidad de datos de entrenamiento.
Para efectuar las corridas de dichos algoritmos utilizaremos el software Weka una herramienta libre para el aprendizaje automático de la información.

Palabras Clave
Minería de datos, Weka, Software Libre, algoritmos de clasificación.

Introducción

[1] La minería de datos consiste en la extracción no trivial de información que reside de manera implícita en los datos. Dicha información era previamente desconocida y podrá resultar útil para algún proceso. En resumen, la minería de datos prepara, sondea y explora los datos para sacar la información oculta en ellos.

Minería de datos abarca todo un conjunto de técnicas enfocadas en la extracción de conocimiento implícito en las bases de datos. Las bases de la minería de datos se encuentran en la inteligencia artificial y en el análisis estadístico. Mediante los modelos extraídos utilizando técnicas de minería de datos se aborda la solución a problemas de predicción, clasificación y segmentación.

Un proceso típico de minería de datos consta de los siguientes pasos generales:

Selección del conjunto de datos: tanto en lo que se refiere a las variables dependientes, como a las variables objetivo, como posiblemente al muestreo de los registros disponibles.
Análisis de las propiedades de los datos: en especial los histogramas, diagramas de dispersión, presencia de valores atípicos y ausencia de datos (valores nulos).
Transformación del conjunto de datos de entrada: en esta etapa se realizará un conjunto de operaciones con la finalidad de preparar los datos de análisis, con el objetivo de adaptarlos para aplicar la técnica de minería de datos que mejor se adapte al problema.
Seleccionar y aplicar la técnica de minería de datos: La elección de la técnica dependerá de la naturaleza del problema a resolver. Para poder implementar la técnica seleccionada, se debe proceder a elegir algún software que facilite el trabajo de aprendizaje automático.
Evaluar los resultados: contrastándolos con un conjunto de datos (datos de entrenamiento) previamente reservados para validar la generalidad del modelo. A continuación se muestra en la Figura 1, como es la proporción de carga de trabajo en cada una de las etapas.

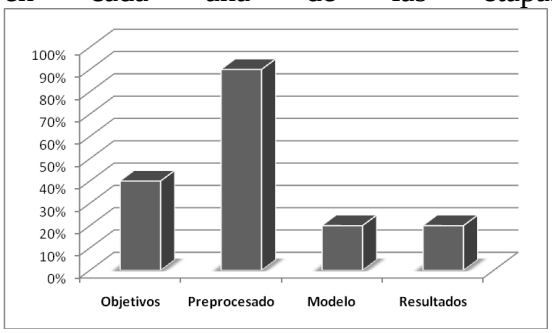

Figura 1. Tiempo estimado en el proceso de minería de datos.

Si el modelo obtenido no superara esta evaluación el proceso se podría repetir desde el principio o, si se considera oportuno, a partir de cualquiera de los pasos anteriores.

Como ya se ha comentado, las técnicas de la minería de datos provienen de la Inteligencia artificial y de la estadística, dichas técnicas, no son más que algoritmos, más o menos sofisticados que se aplican sobre un conjunto de datos para obtener resultados. En la fase de minería de datos, se decide cuál es la tarea a realizar (clasificar, agrupar etc.) y se elige la técnica descriptiva o predictiva que se va a utilizar.

A continuación se muestra en un gráfico con las principales técnicas de minería de datos:

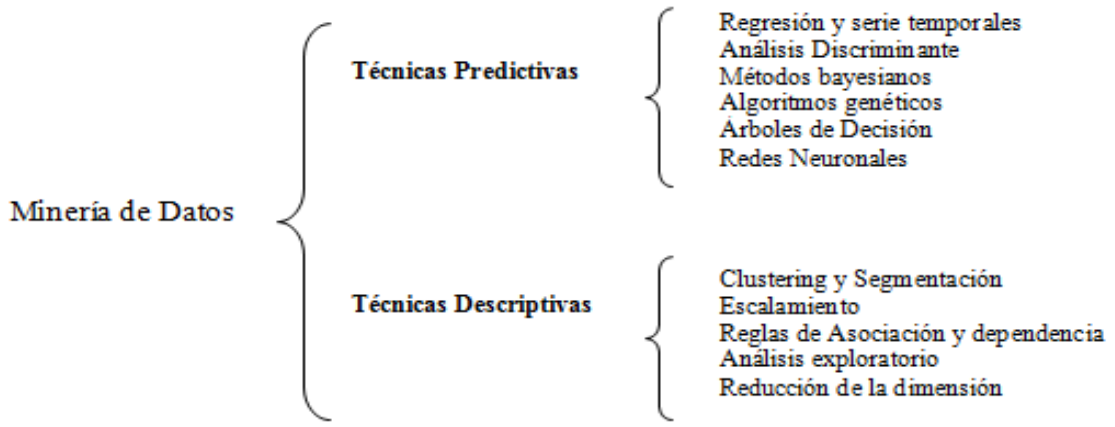

Figura 2. Técnicas de minería de datos.

A continuación se detallan algunas de las técnicas más usadas.

Redes neuronales: Se trata de un sistema de interconexión de neuronas en una red que colabora para producir un estímulo de salida. Algunos ejemplos de red neuronal son: El Perceptrón, El Perceptrón multicapa. Los Mapas Autoorganizados, también conocidos como redes de Kohonen.

Árboles de decisión: Un árbol de decisión es un modelo de predicción utilizado en el ámbito de la inteligencia artificial, dada una base de datos se construyen estos diagramas de construcciones lógicas, muy similares a los sistemas de predicción basados en reglas, que sirven para representar y categorizar una serie de condiciones que suceden de forma sucesiva, para la resolución de un problema. Ejemplos: Algoritmo ID3, Algoritmo C4.5.

Agrupamiento o Clustering: Es un procedimiento de agrupación de una serie de ítems según criterios habitualmente de distancia; se tratará de disponer los vectores de entrada de forma que estén más cercanos aquellos que tengan características comunes. Ejemplos: Algoritmo K-means, Algortimo K-medoids.

Para llevar a cabo el objetivo de este trabajo, aplicaremos minería de datos usando el paquete de software Weka.
Weka es un acrónimo de Waikato Environment for Knowledge Analysis, es un entorno para experimentación de análisis de datos que permite aplicar, analizar y evaluar las técnicas más relevantes de análisis de datos, principalmente las provenientes del aprendizaje automático, sobre cualquier conjunto de datos del usuario.

WEKA se distribuye como software de libre distribución desarrollado en Java.
Está constituido por una serie de paquetes de código abierto con diferentes técnicas de preprocesado, clasificación, agrupamiento, asociación, y visualización, así como facilidades para su aplicación y análisis de prestaciones cuando son aplicadas a los datos de entrada seleccionados.

[2] Las principales herramientas de Weka son:

Simple CLI: la interfaz "Command-Line Interfaz" es simplemente una ventana de comandos java para ejecutar las clases de WEKA.

Explorer: es la opción que permite llevar a cabo la ejecución de los algoritmos de análisis implementados sobre los ficheros de entrada, una ejecución independiente por cada prueba.
Esta es la opción sobre la que se centra la totalidad de esta guía.
Experimenter: esta opción permite definir experimentos más complejos, con objeto de ejecutar uno o varios algoritmos sobre uno o varios conjuntos de datos de entrada, y comparar estadísticamente los resultados.

KnowledgeFlow: esta opción permite llevar a cabo las mismas operaciones del "Explorer", con una configuración totalmente gráfica, inspirada en herramientas de tipo "data-flow" para seleccionar componentes y conectarlos en un proyecto de minería de datos, desde que se cargan los datos, se aplican algoritmos de tratamiento y análisis, hasta el tipo de evaluación deseada.

Los datos de entrada a la herramienta, sobre los que operarán las técnicas implementadas, deben estar codificados en un formato específico, denominado **Attribute-Relation File Format** (extensión "arff"). La herramienta permite cargar los datos en tres soportes: archivo de texto, acceso a una base de datos y acceso a través de internet sobre una dirección URL de un servidor web.
En el caso de los archivos de texto podemos generarlo con cualquier editor de texto, pero al guardarlo debemos modificarle la extensión **".arff"**.
La estructura que debe tener este archivo para poder ser leído por Weka es:

```
% comentarios
@relation NOMBRE_RELACION
@attribute valor1 real
@attribute valor2 real ...
...
@attribute nro1 integer
@attribute nro2 integer
...
@attribute s1 {v1_s1, v2_s1,...vn_s1}
@attribute s2 {v1_s1, v2_s1,...vn_s1}
@data
DATOS
```

Figura 3. Estructura de archivo .arff

Una vez explicado las generalidades del software, nos enfocaremos a realizar pruebas de clasificación enfocándonos en los algoritmos basados en reglas.

[3] El objetivo de las técnicas de clasificación es la asignación de objetos a uno de varios grupos bien definidos.
El proceso general para generar un modelo de clasificación se resume en el siguiente gráfico.

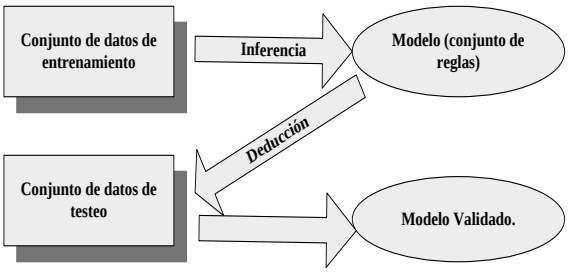

Figura 4. Etapas para la generación de un modelo de clasificación.

Dentro del mundo de reconocimiento de patrones existen dos grandes grupos de familias que enfocan de una manera diferente el problema de la clasificación.

Por un lado está la clasificación no supervisada, trata a la clasificación como el descubrimiento de las clases de un determinado problema. Es decir que contamos con un conjunto de elementos descriptos por un conjunto de características, sin conocer a que clase pertenece cada uno de ellos.

En cambio en la clasificación supervisada enfoca el problema de clasificación de otra manera, es decir parte de un conjunto de elementos descripto por un conjunto de características y conocemos la clase al cual pertenece. A este concepto se lo suele denominar conjunto de datos de entrenamiento o conjunto de aprendizaje.
La clasificación supervisada ha sido aplicada en numerosos ámbitos como el diagnostico de enfermedades, la concesión o rechazo de créditos bancarios, detección de anormalidades en cromosomas etc.

Otro concepto fundamental en el ámbito de los métodos de clasificación son los diversos criterios para la evaluación de los clasificadores. Es decir estimar la bondad de un clasificador, se conoce como proceso de validación, y esto nos permite efectuar una medición sobre la capacidad de predicción del modelo generado a partir de un clasificador.

[4] Una alternativa de verificar o medir la bondad del clasificador es la matriz de confusión.
Una matriz de confusión nos permite visualizar mediante una tabla de contingencia la distribución de errores cometidos por un clasificador.
Esta matriz de confusión para el caso de dos clases tiene la siguiente apariencia, como se puede apreciar en la Tabla.

	Clase Predicha	
Clase real	**Play**	**No Play**
Play	Verdaderos positivos (VP)	Falsos negativos(FN)
No Play	Falsos positivos (FP)	Verdaderos negativos (VN)

Figura 5. Descripción de Matriz de Confusión.

VP (Verdaderos positivos): instancias correctamente reconocidas por el sistema.
FN (Falsos negativos): instancias que son positivas y que el sistema dice que no lo son.
FP (Falsos positivos): instancias que son negativas pero el sistema dice que no lo es.
VN (Verdaderos negativos): instancias que son negativas y correctamente reconocidas como tales.
Suponiendo que N es el número del conjunto de datos de entrenamiento, N=VP+FN+FP+VN.
El número de instancias clasificadas correctamente es la suma de la diagonal de la matriz y el resto están clasificadas de forma incorrecta.
Otra de las maneras de validar más frecuente es basarnos en la tasa de error y la tasa de acierto de un clasificador. Estas tasas se calculan de la siguiente manera:

Tasa de error=FP+FN/N

Tasa de acierto: VP+VN/N

Hay en determinadas situaciones en los que los errores no se valoran por igual, a veces se prefiere extraer más instancias asumiendo que alguno de los propuestos no lo sean, por lo que el costo de un falso positivo es menor que el de una falso negativo. Bajo estas circunstancias es posible incorporar los costos por cada tipo de error. De esta forma se puede integrar en una matriz los beneficios y costos donde Bs representan beneficios y Cs representan costos.

	Clase Predicha	
Clase real	**Play**	**No Play**
Play	Bvp	Cfn
No Play	Cfp	Bvn

Figura 6. Matriz de Costos-Beneficios

La integración de una matriz de confusión y una matriz de costos-beneficios nos brinda las siguientes valoraciones del modelo obtenido.

beneficio=VP. Bvp+VN. Bvn

costo=FP. Cfp+ FN. Cfn

Existen otras aproximaciones para medir la bondad de un clasificador, que serán explicados con más detalle en las pruebas efectuadas.
Dentro de las técnicas de clasificación encontramos: los árboles de clasificación, análisis discriminante, redes neuronales, máquinas de soporte vectorial, redes de bayes y la clasificación basada en reglas que es la técnica que se enfoca nuestro trabajo.

La clasificación en base a reglas es una técnica de clasificación de objetos en base a un conjunto de R reglas del tipo "Si….Entonces….."
Ejemplo:
Si Outlook=rainy y temperature=71.0 y humidity=91.0 y windy=true→ NO JUGAR

La primera parte de la regla es el antecedente y la segunda el consecuente.
Estas reglas deben cumplir una serie de propiedades:
Mutuamente Exclusivas (ME): es decir las "r" de R son (ME) si no se dispara más de una r para un determinado registro.
Exhaustivas: R tiene una cobertura exclusiva si posee una "r" para cada combinación de valores de los atributos. Esto asegura que toda instancia es cubierta por al menos una r de R.
La construcción de clasificadores en base a reglas puede hacerse mediante métodos directo, es decir se extraen reglas desde el conjunto de datos de entrenamiento o por medio de métodos indirectos como lo puede ser por medio de la generación de un árbol de clasificación.
En este trabajo las reglas se generarán a partir del conjunto de datos de entrenamiento.
Una vez generadas las reglas es fundamental medir la calidad de mismas, esto se hace por medio de las siguientes mediciones, la cobertura y la exactitud.

cobertura(r)= |A| / |D|

exactitud(r)= |A Ω y| / |A|

Donde:
|A|: es la cantidad de instancias de un conjunto de entrenamiento D, que satisface la precondición.
|A Ω y|: cantidad de instancias del conjunto de entrenamiento que satisface la precondición y el consecuente.
|D|: representa el número total del conjunto de entrenamiento.

Elementos del Trabajo y metodología

En este trabajo se ha utilizado la herramienta Weka y los datos sobre los cuales se efectuaron las pruebas, han sido extraídos de
Antes de empezar a detallar la metodología de trabajo, partimos de una hipótesis que es la que vamos a validar.
Con las técnicas de clasificación llegamos a obtener un modelo con la finalidad de predecir futuros comportamientos ante nuevos casos. Suponemos que mientras más datos de entrenamiento disponemos, la confiabilidad del modelo obtenido es mayor.
Para poder efectuar las pruebas hemos tomado un archivo de datos "weather.arff". A este mismo archivo lo hemos cargado con 300, 1420 y 2000 instancias.
La estructura del archivo es la que se muestra a continuación en la figura.

Esto que se visualiza es la estructura del archivo y las instancias cargadas.
Esta visualización es posible en la pestaña de Preprocesado y no sólo facilita la navegación sino también la edición en caso de que sea necesario.

A continuación detallaremos el nombre de los algoritmos de clasificación aplicados y los resultados referentes a la confiablidad del modelo obtenido para los archivos cargados con 300, 1420 y 2000 instancias.

Al aplicar un algoritmo de clasificación en Weka tenemos un conjunto de tests:

[2] **Use training set**: Con esta opción Weka entrenará el método con todos los datos disponibles y luego lo aplicará otra vez sobre los mismos.

Supplied test set: Marcando esta opción tendremos la oportunidad de seleccionar, un fichero de datos con el que se probará el clasificador obtenido con el método de clasificación usado y los datos iniciales.

Cross-validation: Weka realizará una validación cruzada estratificada del número de particiones dado (Folds). La validación cruzada consiste en: dado un número n se divide los datos en n partes y, por cada parte, se construye el clasificador con las n−1 partes restantes y se prueba con esa. Así por cada una de las "n" particiones.

Una validación-cruzada es estratificada cuando cada una de las partes conserva las propiedades de la muestra original (porcentaje de elementos de cada clase).

Percentage split: se evalúa la calidad del clasificador según lo bien que clasifique un porcentaje de los datos que se reserva para test.

En nuestro caso el test elegido para realizar todas las pruebas ha sido Use training set.
En base a los resultados obtenidos hemos establecido un ranking teniendo en cuenta cual de los algoritmos de clasificación basados en reglas, es más confiable.
Se presentan en detalle los resultados de las pruebas realizadas.
Resultados obtenidos con el archivo cargado con **350** instancias.

Algortimo	Instancias bien clasificadas (%)	Instancias mal clasificadas(%)	Kappa statistic	Error absoluto
rules.ZeroR	61%	39%	0	0.47
rules.OneR	87.71%	12.28%	0.72	0.12
rules.PART	99%	0.57%	0.98	0.011
rules.NNge	100%	0%	1	0
rules.ConjunctiveRule	82%	18%	0.65	0.24
rules.Ridor	99%	0.57%	0.98	0.005
rules.DecisionTable	99%	0.28%	0.99	0.005
rules.JRip	99%	0.28%	0.99	0.005

Tabla 1. Resultados de la ejecución de algoritmos (350 instancias)

En la tabla detallada anteriormente el porcentaje de instancias correctamente y mal clasificadas.
El estadístico kappa mide la coincidencia de la predicción con la clase real (1.0 significa que ha habido coincidencia absoluta).
La última columna de la tabla arroja el resultado del nivel de error generado del modelo al haber aplicado el algoritmo.

Algoritmo: rules.NNge.

En el siguiente gráfico se visualiza el modelo obtenido, por medio de un conjunto de reglas. En este caso al aplicar el algoritmo NNge, se han generado 7 reglas.

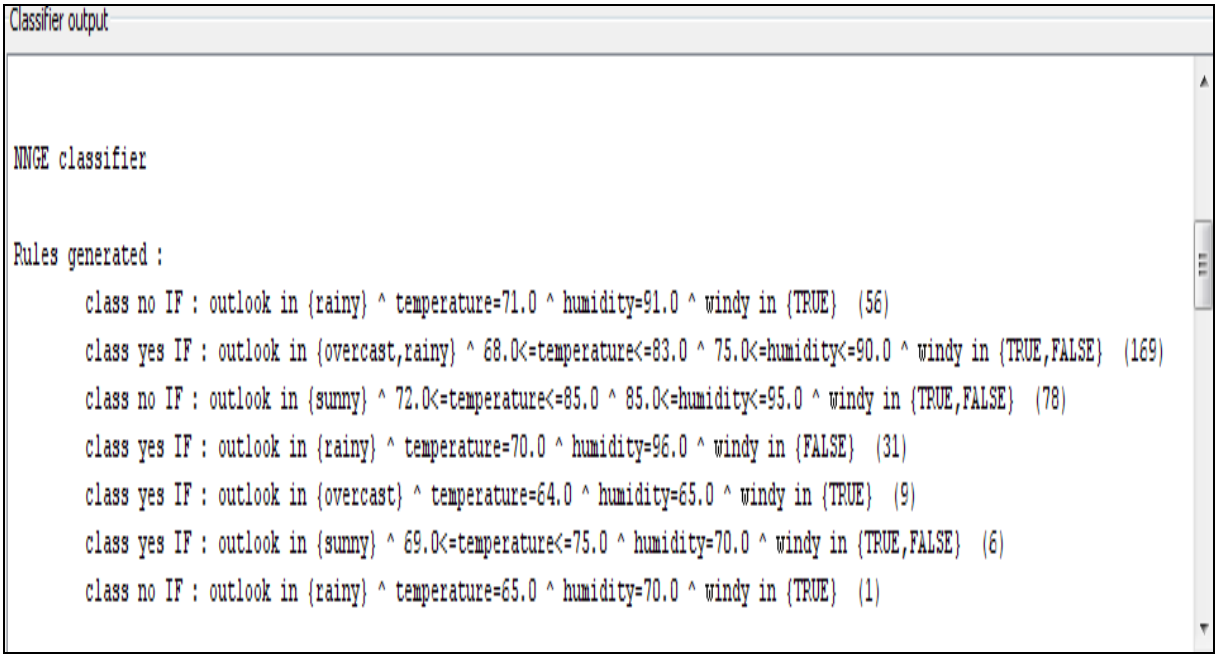

Figura 7. Ventana que visualiza el conjunto de reglas generadas.

Además al ejecutar el algoritmo la ventana del clasificador nos proporciona la siguiente información relacionada con el modelo obtenido.

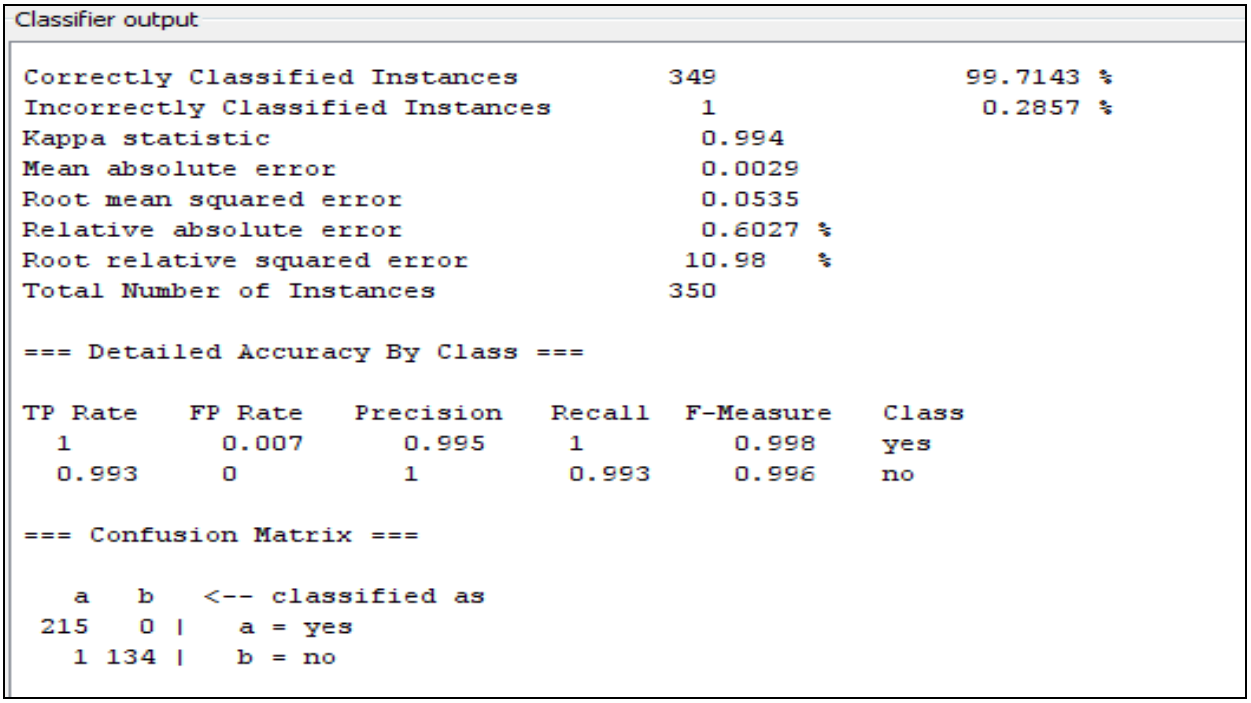

Figura 8. Ventana que visualiza información del modelo generado.

Los parámetros de exactitud del modelo son:

TP (True Positive Rate): es la proporción de ejemplos que fueron clasificados como clase *x*, de entre todos los ejemplos que de verdad tienen clase *x*, es decir, qué cantidad de la clase ha sido capturada. En la matriz de confusión, es el valor del elemento de la diagonal dividido por la suma de la fila relevante.

FP (False Positive Rate): es la proporción de ejemplos que fueron clasificados como

clase *x*, pero en realidad pertenecen a otra clase, de entre todos los ejemplos que no tienen clase *x*. En la matriz de confusión, es la suma de la columna menos el valor del elemento de la diagonal dividido por la suma de las filas de las otras clases.

Cobertura (Recall): la cobertura mide la proporción de términos correctamente reconocidos respecto al total de términos reales, dicho de otro modo, mide en qué grado están todos los que son.

cobertura = VP/VP+VN

Precision: La precisión, en cambio, mide el número de términos correctamente reconocidos respecto al total de términos predichos, sean estos verdaderos o falsos términos. En este caso, la precisión está midiendo la pureza o el grado en que son todos los que están.

precisión = VP/VP+FP

La cobertura y la precisión mantienen una relación inversa, es decir cuando aumenta la cobertura del modelo generado disminuye la precisión y viceversa cuando disminuye la cobertura aumenta la precisión.
De manera tal que se obtiene una cobertura total, a costo de una precisión nula.
También existe otra medida **F-measure** para caracterizar con único valor la bondad de un clasificador o algoritmo. La formula de esta medida está establecida como:

F=2*Precision*Cobertura/(Precision+ Cobertura)

Resultados obtenidos con el archivo cargado con **1420** instancias.

Algortimo	Instancias bien clasificadas (%)	Instancias mal clasificadas(%)	Kappa statistic	Error absoluto
rules.ZeroR	60%	40%	0	0.47
rules.OneR	92%	8%	0.82	0.08
rules.PART	100%	0%	1	0
rules.NNge	100%	0%	1	0
rules.ConjunctiveRule	82%	18%	0.65	0.25
rules.Ridor	100%	0%	1	0
rules.DecisionTable	100%	0%	1	0
rules.JRip	100%	0%	1	0

Tabla 2. Resultados de la ejecución de algoritmos (1000 instancias)

En cuanto al error cometido, Weka ofrece una representación gráfica de los errores que comete el clasificador o algoritmo. Las instancias clasificadas correctamente se representan por cruces y las erróneas por cuadrados. Cada color identifica la clase a la que pertenece la instancia.
Weka ofrece la posibilidad de ver gráficamente en qué atributo comete más error o menos (por ejemplo cuánto se equivoca el algoritmo aplicado si estudiamos si las instancias tienen o no humedad, o tienen o no viento).

Figura 8. Vista gráfica de los errores de clasificación, del algoritmo rulesNNge.

En este caso al visualizar este gráfico relacionado con el error de clasificación al aplicar el algoritmo NNge. Podemos inferir si estudiamos las instancias para determinar si se juega o no tomando como factor el tiempo, observamos que el clasificador es altamente confiable ya que vemos muy pocas instancias mal clasificadas.
En el gráfico de visualización de clasificación se pueden ver las instancias bien clasificadas están representadas por una cruz y las instancias mal clasificadas están representados por un cuadrado.

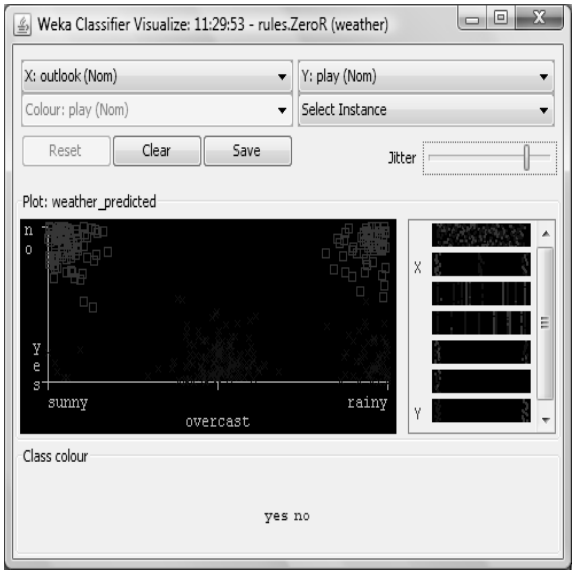

Figura 9. Vista gráfica de los errores de clasificación, del algoritmo rulesZeroR.

En este segundo gráfico de visualización del error de clasificación hemos aplicado el algoritmo rules.ZeroR y se puede apreciar para las mismas coordenadas que existe una mayor proporción de instancias incorrectas es decir mal clasificadas.
Resultados obtenidos con el archivo cargado con **2000** instancias.

Algortimo	Instancias bien clasificadas.(%)	Instancias mal clasificadas(%)	Kappa statistic	Error absoluto
rules.ZeroR	60%	40%	0	0.47
rules.OneR	92%	8%	0.82	0.08
rules.PART	100%	0%	1	0
rules.NNge	100%	0%	1	0
rules.ConjunctiveRule	80%	20%	0	0.47
rules.Ridor	100%	0%	1	0
rules.DecisionTable	100%	0%	0	0.46
rules.JRip	100%	0%	1	0

Tabla 3. Resultados de la ejecución de algoritmos (2000 instancias)

A continuación se reflejan en gráfico cual es el nivel de variación relacionado con confiabilidad de los algoritmos de clasificación.

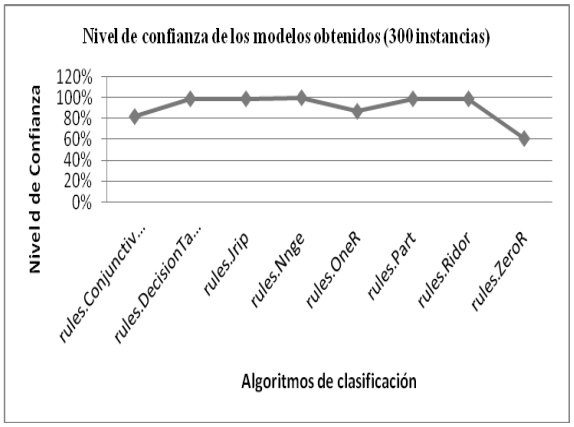

Figura 10. Gráfico que visualiza el nivel de confianza de los modelos, con 300 instancias.

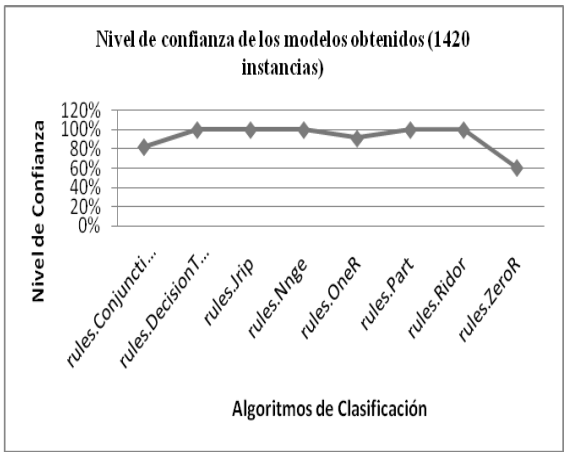

Figura 11. Gráfico que visualiza el nivel de confianza de los modelos, con 1240 instancias.

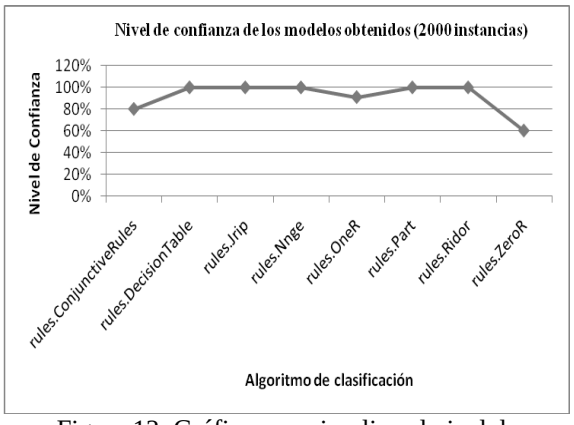

Figura 12. Gráfico que visualiza el nivel de confianza de los modelos, con 2000 instancias.

Resultados

Según los resultados obtenidos de las pruebas, la cantidad de instancias no influye de manera significativa en la bondad de los modelos generados por la aplicación de los algoritmos de clasificación basada en reglas.

Si analizamos las tablas y los gráficos se puede evidenciar que con muy pocas o más instancias, el 62% de los algoritmos de clasificación aplicados proporcionan una confiablidad exacta del modelo obtenido.
Es decir que nuestra hipótesis formulada al inicio del trabajo queda invalidada.

Discusión

En este trabajo se ha enfocado el análisis de rendimiento de las técnicas de clasificación basadas en reglas que implementa el software Weka. Y se ha aplicado sobre un conjunto de datos nominales y no numéricos, por lo que sería interesante investigar que sucede con el nivel de confianza para conjunto de datos de entrenamiento de esta naturaleza.

Es importante destacar que en el caso de que aumenten las instancias en un valor muy superior al considerado en la segunda etapa de pruebas, no se garantiza que la confiabilidad se mantenga o crezca.

Existen otros trabajos que han aplicado el rendimiento de algoritmos referidos a otras técnicas de data mining como clustering y asociación. Además han realizado pruebas para diferente naturaleza de conjunto de entrenamiento y usando diferentes test.

Conclusión

En resumen, data mining se presenta como una tecnología innovadora, que ofrece una serie de beneficios: por un lado, resulta un buen punto de encuentro entre los investigadores y las personas de negocios; por otro, ahorra grandes cantidades de dinero a una empresa y abre nuevas oportunidades de negocios. Además, no hay duda de que trabajar con esta tecnología implica cuidar un sin número de detalles debido a que el producto final involucra "toma de decisiones".

En este trabajo se ha se ha presentado una alternativa de software de minería de datos Weka, que es una herramienta libre y muy interesante a la hora de aplicar diversas técnicas de minería de datos.

Además es importante destacar, que con este trabajo se han detectado cuales de los algoritmos de clasificación basados en reglas tienen una confiablidad interesante, para poder aplicarlo en determinadas temáticas.

A la hora de seleccionar un algoritmo de clasificación basada en reglas este trabajo concluye que el que tiene menos confianza es el rules.ZeroR.

Referencias

[1] López Molina José Manuel, Herrero José García, "Técnicas de Análisis de Datos. Aplicaciones prácticas utilizando Microsoft Excel y Weka", 2006

[2] Capitulo 1. Técnica de Análisis de Datos en Weka.

[3] P. Tan, M. Steinach, V. Kumar, "Introduction to Data Mining", 2006

[4] Jordi Porta Zamorano, "Técnicas cuantitativas para la extracción de términos en un corpus", Universidad autónoma de Madrid, 2006

[5] Pérez López Cesar, González Santín Daniel; "Minería de Datos. Técnicas y Herramientas", Editorial Thompson, 2007

[6] Sierra Araujo Basilio, Arbelaitz Olatz, Armañanzas Rubén, Arruti Andone, Bahamonde Antonio, "Aprendizaje automático: conceptos básicos y avanzados: aspectos prácticos utilizando el software WEKA", Pearson, 2006

[7] María García Jiménez, Aránzazu Álvarez Sierra, "Análisis de Datos en Weka. Pruebas de selectividad.", http://www.it.uc3m.es/jvillena/irc/practicas/06-07/28.pdf

[8] Dapozo Gladys, Porcel Eduardo, López María, Bogado Verónica, Bargiela Roberto, "Aplicación de minería de datos con una herramienta de software libre en la evaluación de rendimiento académico de los alumnos de la carrera de Sistemas de la FACENA-UNNE", Anales del VII Workshop de Investigadores en Ciencias de la Computación, 2006

[9] Chesñevar Carlos Iván, Data mining y aprendizaje automático. Evaluación de lo aprendido/Credibilidad, 2007

Datos de Contacto:
Cynthia Lorena Corso.Universidad Tecnológica Nacional.5009. E-mail: cynthia@bbs.frc.utn.edu.ar