UNIVERSIDAD AUTÓNOMA DEL ESTADO DE HIDALGO
INSTITUTO DE CIENCIAS BÁSICAS E INGENIERÍA
LICENCIATURA EN SISTEMAS COMPUTACIONALES

PROYECTO FINAL
MINERÍA DE DATOS

Alumno:
David Zahid Jiménez Grez.
Asesor:
Dra. Anilú Franco Arcega.

Fecha: 26 de Mayo de 2014

Índice general

Índice de figuras

Índice de tablas

Resumen

Este proyecto en primer lugar describe los métodos de clasificación supervisada utilizados, así como la descripción de la plataforma de software para minería de datos conocida como Weka, la cual se utilizó para el procesamiento de los conjuntos de datos.

A continuación expone cada uno de los conjuntos de datos utilizados, así como las instancias y atributos con las que éstos cuentan, cabe mencionar que los métodos de clasificación supervisada utilizados fueron (Naïve Bayes, Árboles de Decisión, Redes Neuronales, K-Vecinos y Máquinas de Vectores de Soporte.), además, aludir que para la evaluación se utilizó la forma cross-validation (Validación Cruzada). Aunado a esto, por medio de gráficas se muestran los resultados de los métodos de clasificación supervisada, y en éstas se observa el porcentaje de clasificación y tiempo de cada uno de éstos.

Seguidamente, por cada conjunto de datos que se analizó, se concluye cuál de los métodos obtiene el mejor resultado, tomando en cuenta el mejor porcentaje de clasificación correcta y el que logró el menor tiempo de procesamiento.

Finalmente, este documento presenta la conclusión de todo el trabajo y una conclusión general del análisis de los conjuntos de datos.

Abstract

This project in first place describes supervised classification methods used as well as the description of the software platform for data mining known as Weka, which was used to process data sets.

Then exposes each of the used data sets and the instances and attributes with which they have, we should mention that the supervised classification methods used were (Naïve Bayes, Decision Trees, Neural Networks, K-Neighbours and Machines Support Vector.) also allude that the cross-validation assessment form was used. Added to this, by means of graphics supervised classification methods results are shown, and the percentage of these classification and each time they are observed.

Followed, for each data set analyzed, we conclude which method gets the best result, taking into account the best percentage of correct classification and which has the lowest processing time.

Finally, this paper presents the conclusion of all work and a general conclusion of the analysis of data sets.

Introducción

La minería de datos es un conjunto de técnicas que nos ayudan a explorar información valiosa de un conjunto de datos con el fin de extraer información útil para la toma de decisiones. La minería de datos principalmente se usa para describir y predecir; Existen dos modelos que son utilizados dentro de la minería de datos, el modelo descriptivo y el predictivo. La minería de datos puede utilizarse en distintas áreas, algunas de las aplicaciones en general donde se presta la minería de datos son:

- Aplicaciones financieras y banca.
- Aplicaciones de mercado, distribución y comercio.
- Aplicaciones de seguro y salud privada.
- Aplicaciones de educación.
- Aplicaciones en procesos industriales.
- Aplicaciones en medicina.
- Aplicaciones en biología, bioingeniería.

La minería de datos es una disciplina que abarca muchos aspectos de nuestra vida real.
Modelo descriptivo:

- Agrupación.

Modelo preventivo:

- Clasificación.
- Regresión.

En este tipo de clasificación se tiene un conjunto de ejemplos, los cuales tienen asociados un atributo llamado la clase de cada ejemplo, tendremos entonces algún mecanismo de entrenamiento en base a este conjunto, para permitirnos distinguir las clases sobre ejemplos de prueba que proporcionemos.

Este trabajo se realizo para la observación de la eficiencia de los métodos, tanto en tiempo como en calidad de clasificación, en cada conjunto de entrenamiento.

Capítulo 1

Algoritmos de Minería de Datos

Uno de los aspectos más relevantes de la minería de datos es la clasificación de objetos, un objeto en este experimento puede ser tomado como una instancia dentro de un conjunto de datos que representa el conjunto de entrenamiento. Existen diversas métodos para clasificar este objeto, en este capítulo se describen breve mente los métodos utilizados en Weka para procesar los conjunto de datos, entre ellos podemos mencionar el método Naïve Bayes que es aquel que nos permite clasificar un objeto dependiendo con base en sus atributos discretos o continuos para los cuales utiliza formulas diferentes, también está el método de árboles que clasifica un objeto con base en el árbol de decisión que construye y éste va evaluando cada atributo y descartando los que no entran dentro de sus ramas. Un método muy utilizado en la inteligencia artificial es el de Perceptron Simple que en general arroja muy buenos resultado pero resulta más difícil de interpretar en la minería de datos. Otro método que resulta bastante interesante y que fue de los primeros en desarrollarse es K-vecinos este método calcula la distancia de un objeto a otro y clasifica dicho objeto dentro de la clase de aquellos a los que más se parece o con quienes tiene menor distancia. El método de Máquinas de Vectores de Soporte es otro de los método utilizados este método crea hiperplanos como fronteras de decisión [4] .

1.1. Método Naïve Bayes (NB)

El método bayesiano es uno de los más utilizados, se utiliza cuando se tiene incertidumbre o se requiere verificar si un objeto pertenece o no a una clase, ya que permite calcular la probabilidad asociada a dicha hipótesis, dentro del método **NB** se dice que todos los atributos son independientes, conocido el valor de la clase, en este teorema se establece que no importan los atributos que tengo, aun así son válidos para cada una de las clases que

se tiene, se establece: "la probabilidad de que un atributo pertenezca a una clase, por la probabilidad de cada atributo, dadas, dados los valores de cada atributo, ej. Si el máximo pertenece a esa clase, es precisamente esa clase la que se devuelve". Si el atributo es discreto la probabilidad se obtiene por estimación de máxima verosimilitud[4].

$$\frac{P(X_i|pa(X_i))}{n(X_i|pa(X_i))} = \frac{n(X_i, Pa(X_i))}{n(Pa(X_i))} \tag{1.1}$$

Estimación basada en la Ley de Sucesión de LaPlace.

$$P(X_i|pa(X_i)) = \frac{n(X_i, Pa(X_i)) + 1}{n(Pa(X_i)) + |\Omega(X_i)|} \tag{1.2}$$

La cardinalidad es el número de posibles valores que puede tener el atributo evaluado. Si X es continuo el atributo sigue una distribucion normal calcula la media y la desviacion estandar.

$$P(X_i|c)\alpha N(\mu, \sigma) = \frac{1}{\sqrt{2\pi * \sigma}} exp(\frac{(X-\mu)^2}{2\text{ffl}\sigma^2}) \tag{1.3}$$

Para calcular la media y la desviación estándar se utilizan las siguientes formulas:

$$\mu = \frac{1}{n}\sum_{i=1}^{n} a_1 \tag{1.4}$$

$$\sqrt{\sigma^2} = \sqrt{\frac{\sum_{i=1}^{n}(X_i - \mu)^2}{N}} \tag{1.5}$$

1.2. Método de Árboles de decisión (AD)

Un AD es un modelo de predicción.

El algoritmo de **AD** posiblemente es uno de los más utilizados en el aprendizaje automático, ya que cuenta con características que lo hacen destacar entre otros algoritmos, algunas de las cuales se mencionaran a continuación:

- Sencillez de modelo.
- Accesibilidad a diferentes implementaciones.
- Explicación que aporta a la clasificación.
- Rapidez a la hora de clasificar nuevos patrones/objetos.

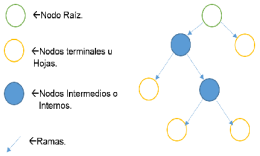

Figura 1.1: Estructura de un Árbol de Decisión

Nodos Internos: Se generan cuando los objetos en el pertenecen a dos o más clases.

Nodos Hoja: Se hacen corresponder con una clase concreta.

Ventajas de los árboles de decisión:

- Aportan una explicación a la clasificación.
- Son capaces de extraer una estructura que representa, el concepto o patrón de comportamiento que hay asociado a la muestra sobre lo que se ha inducido.

Los **AD** nos ayudan a representar la información de manera jerárquica y de esta manera ver en qué clase se encuentra un atributo de una instancia, de tal manera que va buscando el atributo con mayor ganancia y ese es el que se considera para que entre como rama en un nivel del árbol y va descendiendo hasta que llega a clasificar un atributo correctamente, este tipo de método requiere que se conozca cierta información como la ganancia y entropía de cada subconjunto a evaluar, la entropía se refiere al número de instancias que clasifican dentro de cada uno de los valores de la clase. La clasificación final de una instancia depende de las condiciones que se van siguiendo desde la raíz del árbol hasta el final, se requiere evaluar un atributo dentro de todos sus posibles valores y el que tenga la mayor ganancia clasifica, existen formas distintas de clasificar los atributos cuando son numéricos se requiere aplicar un criterio de partición para poder evaluar este tipo de atributos, ya que de esto dependerá en parte que se elabore un buen árbol se deben añadir los hijos resultantes de cada partición las particiones al menos deben separar ejemplos en distintos hijos, con lo que la cordialidad de los nodos irá disminuyendo a medida que se desciende en el árbol. Entre más particiones se tenga los árboles que se creen serán más explícitos y probablemente precisos. Para cada nodo que se coloca en el árbol se debió haber evaluado una atributo especifico y este no debe repetirse más adelante, es decir ya no se puede evaluar en niveles de abajo ya que se estaría repitiendo y seria como si se ciclara, cada hoja nos indica en donde se están clasificando esas instancias, las instancias que aún no se sabe dónde clasifican van bajando en los niveles del árbol. En weka el algoritmo J4.8. nos permite trabajar con con valores continuos para los atributos, separando los posibles resultados en dos ramas. Los árboles son mas pequeños porque cada hoja no cubre una clase

en particular sino una distribución de clases. Particiones realizadas recursivamente, según la estrategia de profundidad-primero (depth-first). Antes de cada partición de datos, el algoritmo considera todas las pruebas posibles que pueden dividir el conjunto de datos y selecciona la prueba que resulta en la mayor ganancia de información o en la mayor proporción de ganancia de información. Cabe resaltar que para cada atributo continuo se realiza una prueba binaria respecto a a los posibles valores que puede tomar el atributo en los datos. Para realizar este método se comienza por elegir un atributo de prueba, esto se hace mediante la aplicación de la fórmula de entropía, sea una función de ganancia de información, es una propiedad estadística que mide como clasifica un atributo X a los objetos del conjunto de entrenamiento (E). Entropía es la efectividad de un atributo para subdividir un conjunto de ejemplos en subconjuntos[4].

$$[E] = info(E) = -\sum_{(f=1)}^{(k)} Pj \log_2 Pj \tag{1.6}$$

Primero por subconjunto o por nodo y también se debe obtener la entropía general de ese conjunto. Una vez que se tiene la entropía se calcula la efectividad.

$$SE(X, C, A_i) = -\sum_{(V_i j \epsilon A_i)} \frac{|\,[A_i(C) = V_i j]\,|}{|X|} I([A_i(C) = V_c i], C) \tag{1.7}$$

Después se obtiene la ganancia.

$$G(|X|, C, A_i) = I(X, C) - E(X, C, A_i) \tag{1.8}$$

El criterio de ganancia selecciona el test que maximice esta ganancia de información (información mutua entre el test X y la clase)

1.3. Método de Redes Neuronales (RD)

Las **RN** tratan de emular en un medio sintético (maquina, computadora, algoritmo, regla) la forma en la que un ser humano llega a una conclusión mediante el uso de **RN**. Este método sirven como un clasificador de clases por lo tanto cuando se tiene un gran número de datos y variables, una base de datos en la cual se tiene la evolución de esas variables y se tiene un gran poder de cálculo se puede aplicar este método de clasificación. Una **RN** tiene la capacidad de aprendizaje esto se realiza mediante el proceso de entrenamiento, este proceso consiste en ir modificando los pesos para cada entrada de las respectivas neuronas en cada iteración hasta encontrar los pesos adecuados, esto hace que se encuentre los pesos para que se cumpla o se verifique el patrón de entrada-salida. En la arquitectura de una **RN** se

requiere saber el número de neuronas, las entradas que se necesitan, cuántas familias o clases se tiene, y la función de activación que coloca a esta neurona en un estado activo 1, existen diferentes funciones de activación como: Hardlim, Satlins, sigmoidal, y otras. La forma más sencilla o básica de una **RN** es el perceptron simple, el cual es una configuración de neuronas las cuales están en una sola línea según lo que requiera el problema[4]. se requiere tener un patrón de entradas y salidas con el cual se va entrenar la **RN**, por ejemplo puede ser la función de conjunción o disyunción, la función básica para el cálculo de las salidas de una red neuronal es, por ejemplo:

$$S = f(\sum_{(i=1)}^{(n)} e * W + O) \tag{1.9}$$

e= entradas W= pesos (se inician aleatoriamente). O=ófsets o vías. Existe también formas de entrenar una **RN** y de encontrar los métodos, uno de ellos es Backpropagation o mejor llamada en español la propagación hacia delante consiste en agregar un incremento, calcular el incremento de los nodos, posteriormente calcular el peso final, que no es más que el peso inicial sumado al incremento del peso. Estas salidas se introducen en la función de activación y se activa o no se compara el resultado con el patrón deseado. Existen diversos criterios para realizar la parada de este algoritmo, uno de ellos es el número de iteraciones que se desea y otro es que ya se haya acabado de clasificar todas las instancias.

1.4. Método de K-Vecinos (KV)

También llamado la regla del vecino más cercano, en un principio este algoritmo se basaba en que un objeto podía pertenecer a la clase a la que más se pareciera, así que la clase más cercana era la etiqueta de clase que se asignaba al objeto nuevo. Una variante de este método es KV (K-Vecinos), en él se asigna la clase mayoritaria entre los k-vecinos más cercanos. Si hay un empate se aplica el criterio donde se asigna al objeto s clasificar la clase que tenga el primer vecino más cercano entre las empatadas. Este método consiste en medir la distancia entre objetos de acuerdo al valor de los atributos numéricos, la distancia se mide de acuerdo a la fórmula que se desee, existen diferentes tipos de distancia, en este algoritmo se mide la distancia de un objeto al resto de los objetos del conjunto de entrenamiento, luego se toman los resultados de los **KN** más cercanos, es decir aquellos cuya distancia a dicho objeto de entrenamiento sea menor, y de acuerdo a la etiqueta de clase que tengan la mayoría de los **KN** más cercanos esa será la clase asignada al nuevo objeto a clasificar[4].

En este método existen diferentes tipos de distancias que se pueden aplicar, como lo son:

La formula para la distancia euclidiana es la siguiente:

$$d(x, y) = \sqrt{\sum_{i=1}^{n}(X_i - Y_i)^2} \tag{1.10}$$

Esta distancia fue la una de las que se utilizaron en el curso y la cual se utilizó para aplicar el método de K-Vecinos sobre los conjuntos de entrenamiento.

1.5. Método de Máquinas de Vectores de Soporte (MVS)

La **MKV** fue diseñada para la clasificación binaria, éste es un modelo predictivo de aprendizaje supervisado, utiliza planos de decisión, es decir define una frontera de decisión para clasificar a los objetos de entrenamiento, para conjuntos linealmente separables. Para esto se calculan los hiperplanos que son las fronteras de decisión entre cada clase, que son positiva o negativa, éste es un método de aprendizaje supervisado. Para clasificar un nuevo objeto de entrenamiento se calcula el tamaño del vector dependiendo de los atributos que se tengan y multiplica el alfa por los respectivos vectores de los objetos de entrenamiento que se toman como vectores de soporte, luego el valor obtenido de la sumatoria de estos vectores se multiplica por el vector de dicho objeto de entrenamiento de los atributos y por su respectiva clase que es positiva o negativa, enseguida se despeja a b para cada vector, luego se obtiene el promedio de las b despejadas y el resultado de esa b se multiplica por el vector de atributos del objeto de entrenamiento tomado como vector de soporte y por la clase. Esto se realiza para la clasificación de un nuevo objeto y si el resultado es negativo el nuevo objeto a clasificar clasifica en la clase de los negativos y si es positivo en los positivos.

Capítulo 2

Weka

Es un software que funciona como una máquina de aprendizaje automático, oficialmente es un Entorno para Análisis del Conocimiento creado originalmente en 1993 como un proyecto en la Universidad de Waikato, Nueva Zelanda.

Weka tiene grandes ventajas ya que es un software de código abierto y está programado en java que es un lenguaje multiplataforma (entre sus múltiples ventajas), y aunque no hay una documentación muy amplia del software resulta bastante intuitivo para el usuario.

Weka posee una extensa colección de algoritmos de Máquinas de conocimiento desarrollados para entre otras aplicaciones para la minería de datos. Además Weka contiene las herramientas necesarias para realizar transformaciones sobre los datos, tareas de clasificación, regresión, clustering, asociación y visualización. Weka está diseñado como una herramienta orientada a la extensibilidad esto en parte se debe al hecho está programada en java lo que facilita esta característica en el software[3].

Para su funcionamiento Weka recibe un archivo que contiene la base de datos que será utilizada como el conjunto de entrenamiento para la máquina de aprendizaje, el formato que deben tener todos los archivos que son introducidos en Weka es .arff, la estructura de este formato está compuesta por tres partes:

La declaración *@relation*

El nombre de la relación se define como la primera línea en el archivo ARFF . El formato es:

@relation <nombre-relación>

donde <nombre-relacion>es una cadena de textos. La cadena no debe contener espacios.

La declaración *@attribute*

Cada atributo en el conjunto de datos tiene su propia declaración de *@attribute* que define de forma exclusiva el nombre de ese atributo y su tipo de datos. El orden de los atributos se declaran indicando la posición de la

columna en la sección de datos del archivo. El formato para la declaración del *@attribute* es:

@attribute <tipoDeDatos><nombre-atributo>

donde el <nombre-atributo>debe comenzar con un carácter alfabético. Si los espacios se van a incluir en el nombre y luego el nombre completo debe ser citado.

Con <tipoDeDatos>indicaremos el tipo de dato para este atributo (o columna) que puede ser:

- string (texto).
- date [<date-format>] (fecha). En <date-format>indicaremos el formato de la fecha, que será del tipo ƨyyy-MM-dd'T'HH:mm:ss".
- numeric (numérico)
- <nominal-specification>. Estos son tipos de datos definidos por nosotros mismos.

La declaración *@data*

En esta sección incluiremos los datos propiamente dichos. Separaremos cada columna por comas y todas filas deberán tener el mismo número de columnas, número que coincide con el de declaraciónes @attribute que añadimos en la sección anterior. Si no disponemos de algún dato, colocaremos un signo de interrogación (?) en su lugar. El separador de decimales tiene que ser obligatoriamente el punto y las cadenas de tipo string tienen que estar entre comillas simples[5].

Explorer

Interfaz básico para usar el conjunto de algoritmos que ofrece WEKA (clasificación, clustering, reglas de asociación, selección de atributos y visualización).

Experimenter

Interfaz gráfico para automatizar baterías de experimentos.

Knowledge Flow

Interfaz gráfico para diseñar flujos y procesos donde se combinen varios componentes para conformar aplicaciones complejas.

Simple CLI

Acceso los algoritmos de WEKA desde un interfaz de línea de comandos. Para realizar esta práctica se empleará el interfaz Explorer (ver presentación PPT), si se hacen las pruebas "manuales", o el interfaz Experimenter, si se desean automatizar las pruebas.

Las funcionalidades del interfaz Explorer se organizan en 6 pestañas:

Preprocess

Carga de los datasets a emplear y procesamiento previo a la aplicación de los algoritmos de aprendizaje Permite cargar datos desde ficheros ARFF, ficheros CSV y bases de datos (mediante JDBC).

Classify

Interfaz de experimentación con algoritmos de clasificación Permite seleccionar un clasificador (botón [Choose]) y configurar sus parámetros (pulsando sobre el nombre del algoritmo)

Permite especificar el método de evaluación (Test options)

El resultado de aplicar el clasificador elegido será puesto a prueba de acuerdo con las opciones que se establecen haciendo clic en el cuadro de Test options.

Hay cuatro modos de prueba:

- *training set*: esta opción evalúa el clasificador sobre el mismo conjunto sobre el que se construye el modelo predictivo para determinar el error, que en este caso se denomina .error de resustitución". Por tanto, esta opción puede proporcionar una estimación demasiado optimista del comportamiento del clasificador, al evaluarlo sobre el mismo conjunto sobre el que se hizo el modelo.

- *Supplied test set*: evaluación sobre conjunto independiente. Esta opción permite cargar un conjunto nuevo de datos. Sobre cada dato se realizará una predicción de clase para contar los errores.

- *Cross-validation*: evaluación con validación cruzada. Esta opción es la más elaborada y costosa. Se realizan tantas evaluaciones como se indica en el parámetro Folds. Se dividen las instancias en tantas carpetas como indica este parámetro y en cada evaluación se toman las instancias de cada carpeta como datos de test, y el resto como datos de entrenamiento para construir el modelo. Los errores calculados son el promedio de todas las ejecuciones.

- *Percentage split*: esta opción divide los datos en dos grupos, de acuerdo con el porcentaje indicado (%). El valor indicado es el porcentaje de instancias para construir el modelo, que a continuación es evaluado sobre las que se han dejado aparte. Cuando el Instancias es suficientemente elevado, esta opción es suficiente para estimar con precisión las prestaciones del clasificador en el dominio[2].

Permite establecer el atributo clase sobre el que realizar el aprendizaje Muestra los resultados del proceso entrenamiento-evaluación

Cluster

Interfaz de experimentación con algoritmos de clustering

Associate

Interfaz de experimentación con algoritmos para aprendizaje de reglas de asociación

Select attributes

Interfaz de experimentación con algoritmos de selección de atributos

Visualize

Interfaz para la visualización de datasets, relaciones entre atributos, etc [6].

Capítulo 3

Resultados Experimentales

3.1. Abalone

La base de datos Abalone es originaria de un estudio titulado "La población biológica de Abalone (especie Haliotis) en Tasmania.", de la división de recursos de la marina y el laboratorio de investigación marina. La base de datos contiene información que predice la edad de Abalone, dadas sus medidas físicas. La edad de Abalone es determinada al cortar la coraza por el cono, creando una mancha y contando el número de anillos utilizando un microscopio para ello; es una actividad aburrida y que consume mucho tiempo [1].

Instancias	Atributos
4177	9

Tabla 3.1: Número de Instancias y Atributos del conjunto de datos de Abalone

Atributo	Tipo	Dominio
sexo	Categórico	M, F, I
longitud	Real	0.45, 0.33, 0.256, ...
Diámetro	Real	0.365, 0.600, 0.666, etc.
Altura	Real	0.4755, 0.420, 0.1366...
Peso completo	Real.	1.2, 0.4678, ...
Peso sin concha	Real.	0.365, 0.444, 0.3200, ...
Peso visceras	Real.	0.4755, 0.45, 0.33, ...
Peso coraza	Real.	0.4755, 0.4005, 0.3233, ...
clase	Numérico.	4,5,10, 19,... etc.

Tabla 3.2: Atributos del conjunto de datos de Abalone

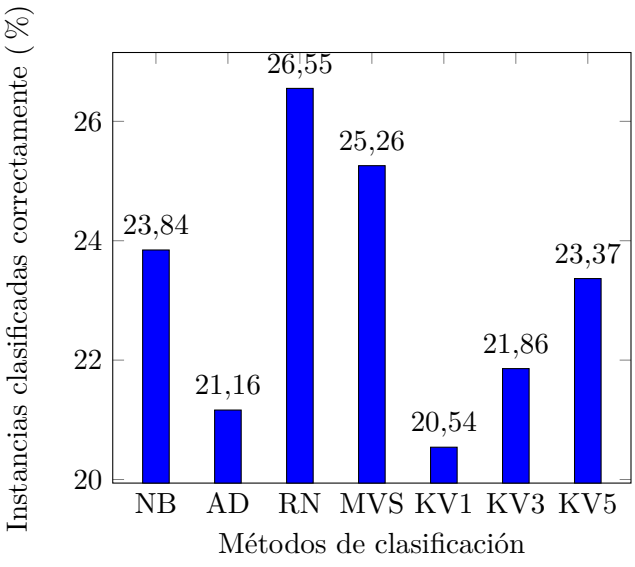

Figura 3.1: Porcentajes de clasificación correcta del conjunto de datos de Abalone

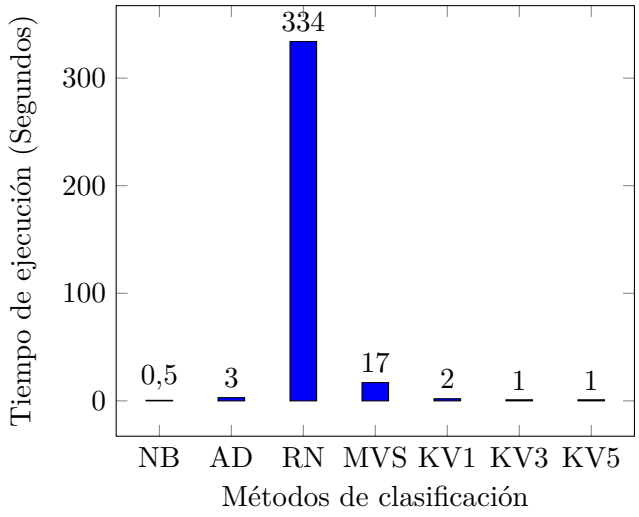

Figura 3.2: Tiempos del procesamiento del conjunto de datos de Abalone

Los resultados generales de la clasificación se muestran en la Figura 3.1 donde se puede ver el porcentaje de clasificación correcta que obtiene cada método de clasificación utilizada. la Figura 3.2 muestra por su parte los resultados del tiempo de procesamiento empleado por cada algoritmo mostrado en segundos. Estos resultados son un porcentaje de la clasificación del conjunto de datos Abalone aplicado a 5 métodos de clasificación. De acuerdo a los métodos que tuvieron un mayor porcentaje de clasificación de instancias, se puede apreciar que para este caso existen dos métodos con un indice alto de clasificación fueron Redes Neuronales y Máquinas de Soporte de Vectores. Por otro lado, es importante considerar el tiempo que tardaron en clasificar estos métodos, como se puede apreciar en la Figura 3.2, el método de Máquinas de Soporte de Vectores se tardó menos tiempo que el método de Redes Neuronales. En los demás métodos de clasificación empleados, se observo que su precisión fue similar, sin embargo su porcentaje de error de clasificación fue mayor a los 2 métodos con mayor porcentaje de clasificación correcta, debido a que tenían errores al momento de clasificar, ésto se observa en las matrices de que se obtuvieron en la aplicación de los métodos. Por lo tanto, para clasificar este conjunto de datos siempre será mejor utilizar el método de Maquinas de Vectores de Soporte debido a que tuvo un porcentaje considerable de clasificación y el tiempo que demoro en clasificar es mínimo en comparación al del otro método.

3.2. Rookpawn

La base de datos de Rey Rook y Rey Pawn (ajedrez) fue donada por Rob Holte y generada y descrita por Alen Shapiro. La base de datos fue suministrada a Holte por Peter Clark del Instituto Turing en Glasgow[1].

Tabla 3.3: Número de Instancias y Atributos Rookpawn

Instancias	Atributos
3196	37

Tabla 3.4: Atributos de Rookpawn

Atributo	Tipo	Dominio
bkblk	Categórico	f,t
bknwy	Categórico	f,t
bkon8	Categórico	f,t
bkona	Categórico	f,t
bkspr	Categórico	f,t
bkxbq	Categórico	f,t
bkxcr	Categórico	f,t
bkxwp	Categórico	f,t
blxwp	Categórico	f,t
bxqsq	Categórico	f,t
cntxt	Categórico	f,t
dsopp	Categórico	f,t
dwipd	Categórico	l,g
hdchk	Categórico	f,t
katri	Categórico	n,w,b
mulch	Categórico	f,t
qxmsq	Categórico	f,t
r2ar8	Categórico	t,f
reskd	Categórico	f,t
reskr	Categórico	f,t
rimmx	Categórico	f,t
rkxwp	Categórico	f,t
rxmsq	Categórico	f,t
simpl	Categórico	f,t
skach	Categórico	f,t
skewr	Categórico	t,f
skrxp	Categórico	f,t
spcop	Categórico	f,t
stlmt	Categórico	f,t
thrsk	Categórico	f,t
wkcti	Categórico	f,t
wkna8	Categórico	f,t
wknck	Categórico	f,t
wkovl	Categórico	t,f
wkpos	Categórico	t,f
wtoeg	Categórico	n,t
clase	Categórico	won,nowin

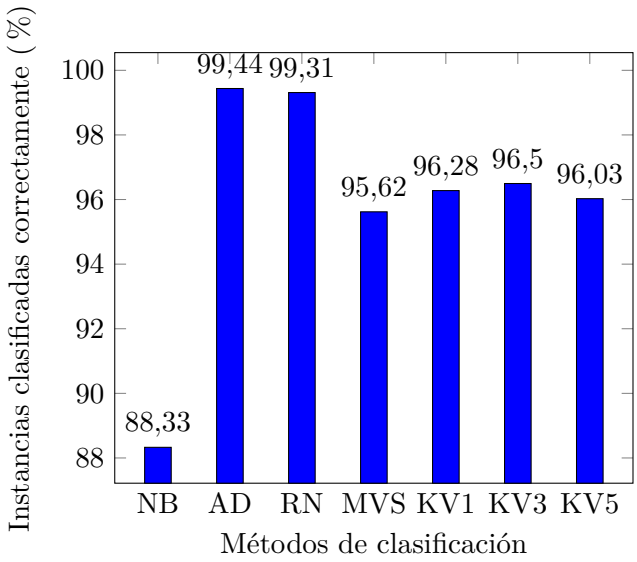

Figura 3.3: Porcentajes de clasificación correcta del conjunto de datos de Rookpawn

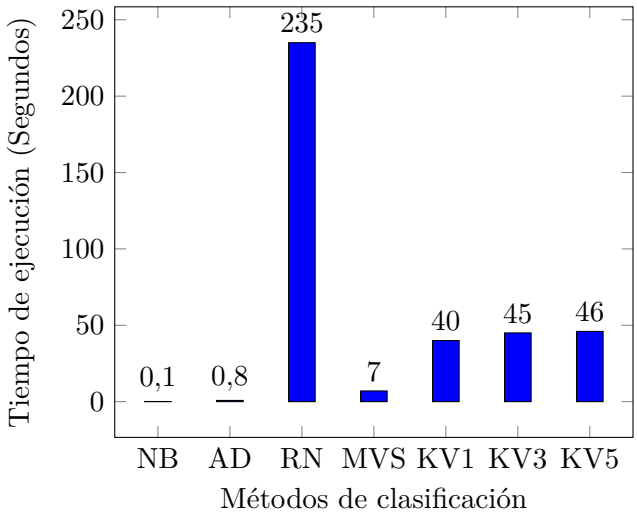

Figura 3.4: Tiempos del procesamiento del conjunto de datos de Rookpawn

Los resultados generales de la clasificación se muestran en la Figura 3.3 donde se puede ver el porcentaje de clasificación correcta que obtiene cada método de clasificación utilizada. la Figura 3.4 muestra por su parte los resultados del tiempo de procesamiento empleado por cada algoritmo mostrado en segundos. Estos resultados son un porcentaje de la clasificación del conjunto de datos Rookpawn aplicado a 5 métodos de clasificación. De

acuerdo a los métodos que tuvieron un mayor porcentaje de clasificación de instancias, se puede apreciar que para este caso existen dos métodos con un indice alto de clasificación fueron Árboles de Decisión y Redes Neuronales. Por otro lado, es importante considerar el tiempo que tardaron en clasificar estos métodos, como se puede apreciar en la Figura 3.4, el método de Árboles de Decisión se tardó menos tiempo que el método de Redes Neuronales. En los demás métodos de clasificación empleados, se observo que su precisión fue similar, sin embargo su porcentaje de error de clasificación fue mayor a los 2 métodos con mayor porcentaje de clasificación correcta, debido a que tenían errores al momento de clasificar, ésto se observa en las matrices de que se obtuvieron en la aplicación de los métodos. Por lo tanto, para clasificar este conjunto de datos siempre será mejor utilizar el método de Árboles de Decisión debido a que tuvo un porcentaje considerable de clasificación y el tiempo que demoro en clasificar es mínimo en comparación al del otro método.

3.3. Segmentación de Piel

El conjunto de datos de la piel es recogida por muestreo aleatorio B, G, R valores de imágenes de caras de diferentes grupos de edad (jóvenes, de mediana y edad), grupos de raza (blanco, negro y asiático), y géneros obtenidos de la base de datos y base de datos FERET PAL. Tamaño de la muestra total de aprendizaje es de 245,057; de los cuales 50,859 son las muestras de piel y 194,198 no son muestras de piel[1].

Tabla 3.5: Número de Instancias y Atributos Segmentación de Piel

Instancias	Atributos
245057	4

Tabla 3.6: Atributos de Segmentación de Piel

Atributo	Tipo	Dominio
R	Entero	0,..255
G	Entero	0,..255
B	Entero	0,..255
Clase	Categórico	1,2

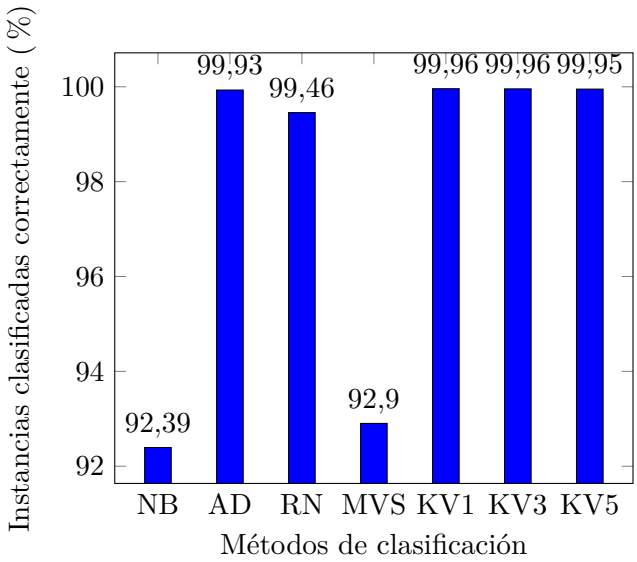

Figura 3.5: Porcentajes de clasificación correcta del conjunto de datos de Segmentación de Piel

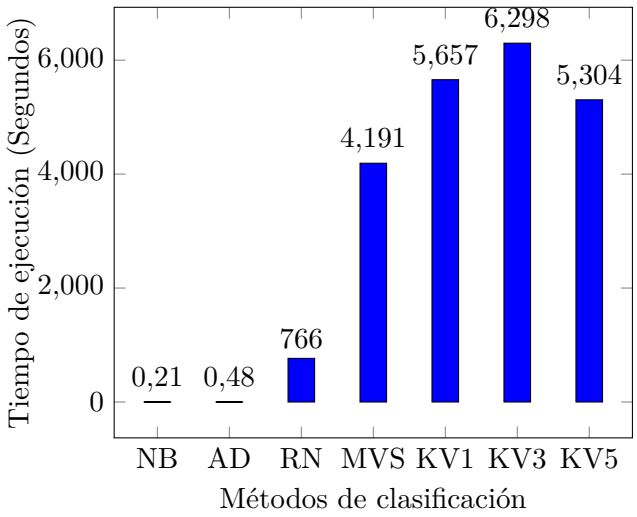

Figura 3.6: Tiempos del procesamiento del conjunto de datos de Segmentación de Piel

Los resultados generales de la clasificación se muestran en la Figura 3.5 donde se puede ver el porcentaje de clasificación correcta que obtiene cada método de clasificación utilizada. la Figura 3.6 muestra por su parte los resultados del tiempo de procesamiento empleado por cada algoritmo mostrado en segundos. Estos resultados son un porcentaje de la clasificación del conjunto de datos de Segmentación de Piel aplicado a 5 métodos de clasificación. De acuerdo a los métodos que tuvieron un mayor porcentaje de clasificación de instancias, se puede apreciar que para este caso existen dos métodos con un indice alto de clasificación fueron Árboles de Decisión y K-Vecinos. Por otro lado, es importante considerar el tiempo que tardaron en clasificar estos métodos, como se puede apreciar en la Figura 3.6, el método de Árboles de Decisión se tardó menos tiempo que el método de K-Vecinos. En los demás métodos de clasificación empleados, se observo que su precisión fue similar, sin embargo su porcentaje de error de clasificación fue mayor a los 2 métodos con mayor porcentaje de clasificación correcta, debido a que tenían errores al momento de clasificar, ésto se observa en las matrices de que se obtuvieron en la aplicación de los métodos. Por lo tanto, para clasificar este conjunto de datos siempre será mejor utilizar el método de Árboles de Decisión debido a que obtuvo un porcentaje considerable de clasificación y el tiempo que demoro en clasificar es mínimo en comparación al del otro método.

3.4. Pokerhand

Cada registro es un ejemplo de una mano con cinco cartas procedentes de una baraja de 52. Cada carta se describe el uso de dos atributos (traje y de la fila), para un total de 10 atributos de predicción. Hay un atributo de clase que describe el "Poker Hand". El orden de las cartas es importante, por lo que hay 480 posibles manos Royal Flush[1].

Tabla 3.7: Número de Instancias y Atributos Pokerhand

Instancias	Atributos
184743	11

Tabla 3.8: Atributos de Pokerhand

Atributo	Tipo	Dominio
juegocarta1	real	1,..4
posicioncarta1	real	1,..13
juegocarta2	real	1,..4
posicioncarta2	real	1,..13
juegocarta3	real	1,..4
posicioncarta3	real	1,..13
juegocarta4	real	1,..4
posicioncarta4	real	1,..13
juegocarta5	real	1,..4
posicioncarta5	real	1,..13
clase	Categórico	0,..9

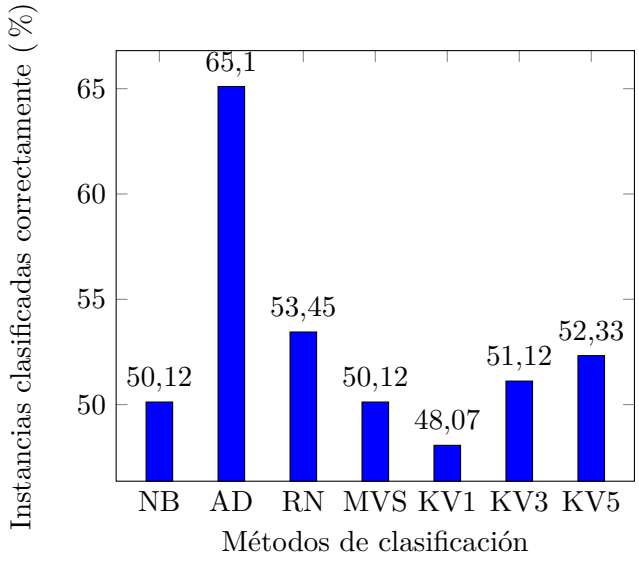

Figura 3.7: Porcentajes de clasificación correcta del conjunto de datos de Pokerhand

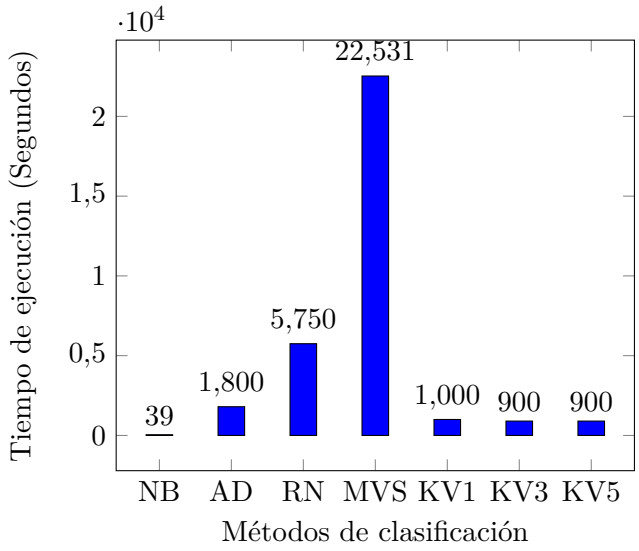

Figura 3.8: Tiempos del procesamiento del conjunto de datos de Pokerhand

Los resultados generales de la clasificación se muestran en la Figura 3.7 donde se puede ver el porcentaje de clasificación correcta que obtiene cada método de clasificación utilizada. la Figura 3.8 muestra por su parte los resultados del tiempo de procesamiento empleado por cada algoritmo mostrado en segundos. Estos resultados son un porcentaje de la clasificación del conjunto de datos Pokerhand aplicado a 5 métodos de clasificación. De

acuerdo a los métodos que tuvieron un mayor porcentaje de clasificación de instancias, se puede apreciar que para este caso existen dos métodos con un indice alto de clasificación fueron Árboles de Decisión y Redes Neuronales. Por otro lado, es importante considerar el tiempo que tardaron en clasificar estos métodos, como se puede apreciar en la Figura 3.8, el método de Árboles de Decisión se tardó menos tiempo que el método de Redes Neuronales. En los demás métodos de clasificación empleados, se observo que su precisión fue similar, sin embargo su porcentaje de error de clasificación fue mayor a los 2 métodos con mayor porcentaje de clasificación correcta, debido a que tenían errores al momento de clasificar, ésto se observa en las matrices de que se obtuvieron en la aplicación de los métodos. Por lo tanto, para clasificar este conjunto de datos siempre será mejor utilizar el método de Árboles de Decisión debido a que obtuvo un porcentaje considerable de clasificación y el tiempo que demoro en clasificar es mínimo en comparación al del otro método.

3.5. Car

INFORMACIÓN ERRONEA (CORREGÍR) El modelo jerárquico de decisión, de la que este conjunto de datos es derivado, se presentó por primera vez en M. Bohanec y Rajkovic V.: La adquisición de conocimiento y explicación de multi-atributo de la toma de decisiones. En el 8 º Seminario Internacional de Expertos sobre Sistemas y sus Aplicaciones, Aviñón, Francia. páginas 59-78, 1988. Dentro del aprendizaje de la máquina, este conjunto de datos se utilizó para la evaluación de Pista, que se ha demostrado ser capaz de reconstruir completamente el modelo jerárquico original. esto, junto con una comparación con C4.5, se presenta en B. Zupan, Bohanec M., Bratko I., Demšar J.: El aprendizaje automático por función de descomposición. ICML-97, Nashville, TN. 1997 (en preparación)[1].

Tabla 3.9: Número de Instancias y Atributos Car

Instancias	Atributos
1728	7

Tabla 3.10: Atributos de Tic tac toe

Atributo	Tipo	Dominio
compra	Categórico	vhigh, high, med, low
mantenimiento	Categórico	vhigh, high, med, low
npuertas	Categórico	2,3,4,5more
npersonas	Categórico	2,4,more
agarraderas	Categórico	small,med,big
seguridad	Categórico	low, med, high
clase	Categórico	unacc,acc,good,vgood

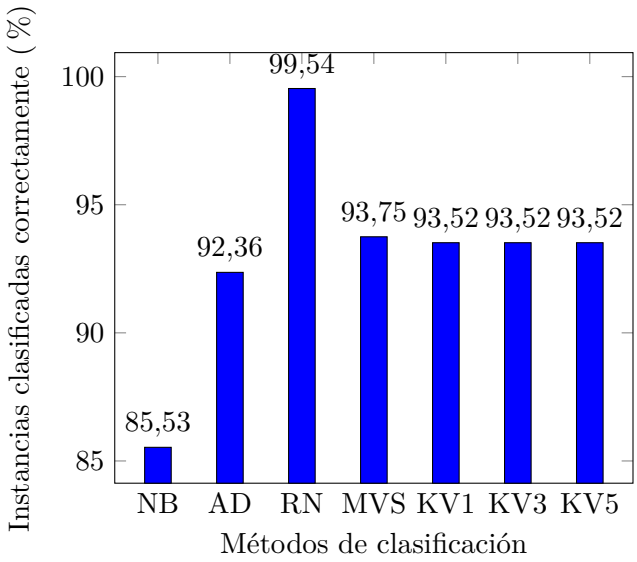

Figura 3.9: Porcentajes de clasificación correcta del conjunto de datos de Car

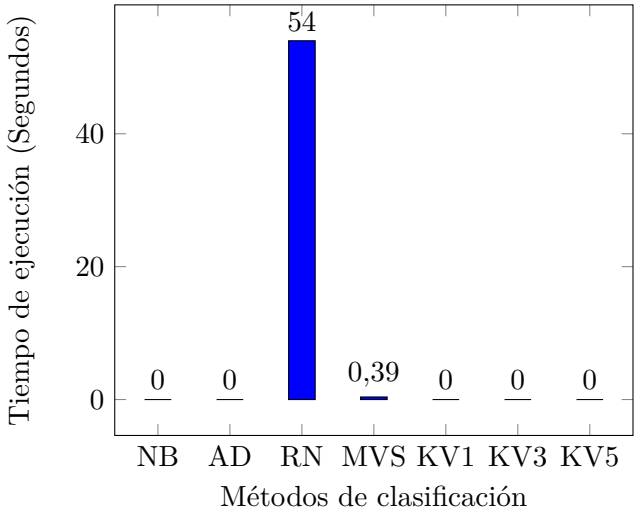

Figura 3.10: Tiempos del procesamiento del conjunto de datos de Car

Los resultados generales de la clasificación se muestran en la Figura 3.9 donde se puede ver el porcentaje de clasificación correcta que obtiene cada método de clasificación utilizada. la Figura 3.10 muestra por su parte los resultados del tiempo de procesamiento empleado por cada algoritmo mostrado en segundos. Estos resultados son un porcentaje de la clasificación del conjunto de datos Car aplicado a 5 métodos de clasificación. De acuerdo a

los métodos que tuvieron un mayor porcentaje de clasificación de instancias, se puede apreciar que para este caso existen dos métodos con un indice alto de clasificación fueron Redes Neuronales y Máquinas de Soporte de Vectores. Por otro lado, es importante considerar el tiempo que tardaron en clasificar estos métodos, como se puede apreciar en la Figura 3.10, el método de Máquinas de Soporte de Vectores se tardó menos tiempo que el método de Redes Neuronales. En los demás métodos de clasificación empleados, se observo que su precisión fue similar, sin embargo su porcentaje de error de clasificación fue mayor a los 2 métodos con mayor porcentaje de clasificación correcta, debido a que tenían errores al momento de clasificar, por otro lado, el método de Naïve Bayes fue el que obtuvo un porcentaje mayor de clasificación incorrecta, ésto se observa en las matrices de que se obtuvieron en la aplicación de los métodos. Por lo tanto, para clasificar este conjunto de datos siempre será mejor utilizar el método de Maquinas de Vectores de Soporte debido a que obtuvo un porcentaje considerable de clasificación y el tiempo que demoro en clasificar es mínimo en comparación al del otro método.

3.6. Seismic-bump

La complejidad de los procesos sísmicos y gran desproporción entre el número de eventos sísmicos de baja energía y el número de fenómenos de alta energía hace que las técnicas estadísticas son insuficientes para predecir la peligrosidad sísmica . Por lo tanto , es esencial para la búsqueda de nuevas oportunidades de una mejor predicción de peligros, también utilizando métodos de aprendizaje automático . En las técnicas de agrupamiento de datos de evaluación de la peligrosidad sísmica se puede aplicar (Lesniak A. , Isakow Z. : . . Clustering El espacio-tiempo de los eventos sísmicos y evaluación hzard en la mina de carbón Zabrze - Bielszowice , Polonia Int. Journal of Rock Mechanics y Ciencias de la Minería , de 46 (5), 2009 , 918-928) , y para la predicción de temblores sísmicos Redes Neuronales artificiales se utilizan (Kabiesz , J. : . Efecto de la forma de los datos sobre la calidad de la mina de la previsión de riesgos temblores usando Redes Neuronales geotécnica y geológica Ingeniería, 24 (5), 2005, 1131-1147).En la mayoría de las aplicaciones, los resultados obtenidos por los métodos mencionados se presentan en forma de dos estados que se interpretan como « peligrosos y « no peligrosos.Distribución desequilibrada de positivo (.estado peligroso") y negativo (.estado no peligrosos") ejemplos es un problema grave en la predicción del riesgo sísmico. Actualmente se utilizan métodos son todavía insuficientes para lograr una buena sensibilidad y especificidad de las predicciones. Los datos presentados conjunto se caracterizan por una distribución desequilibrada de los ejemplos positivos y negativos. En el conjunto de datos no están a sólo 170 ejemplos positivos que representan la clase 1[1].

Tabla 3.11: Número de Instancias y Atributos de Seismic bump

Instancias	Atributos
2584	19

Tabla 3.12: Atributos de Seismicbump

Atributo	Tipo	Dominio
Sísmico	Categórico	a,b,c,d
Sísmicoacustico	Categórico	a,b,c,d
Cambio	Categórico	W, N
Genergia	Real	48,72,-30,448,400,...
Pulsosg	Real	48,72,-30,448,400,...
Peligrog	Categórico	a,b,c,d
Ngolpes	Real	48,72,-30,448,400,...
Ngolpes2	Real	48,72,-30,448,400,...
Ngolpes3	Real	48,72,-30,448,400,...
Ngolpes4	Real	48,72,-30,448,400,...
Ngolpes5	Real	48,72,-30,448,400,...
Ngolpes6	Real	48,72,-30,448,400,...
Ngolpes7	Real	48,72,-30,448,400,...
Ngolpes89	Real	48,72,-30,448,400,...
Energía	Real	48,72,-30,448,400,...
Maxenergia	Real	48,72,-30,448,400,...
Clases	Categórico	1,0

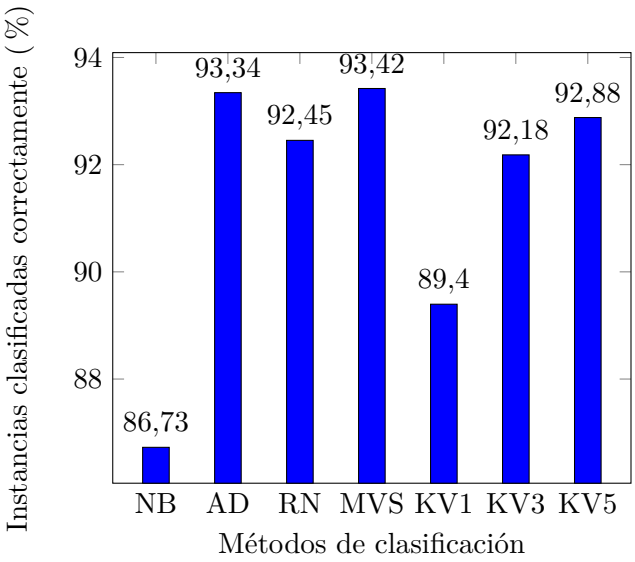

Figura 3.11: Porcentajes de clasificación correcta del conjunto de datos de Seismic-bump

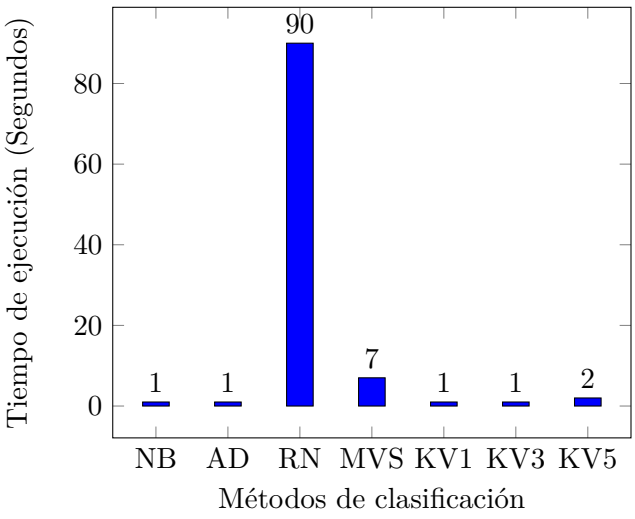

Figura 3.12: Tiempos del procesamiento del conjunto de datos de Seismic-bump

Los resultados generales de la clasificación se muestran en la Figura 3.11 donde se puede ver el porcentaje de clasificación correcta que obtiene cada método de clasificación utilizada. la Figura 3.12 muestra por su parte los resultados del tiempo de procesamiento empleado por cada algoritmo mostrado en segundos. Estos resultados son un porcentaje de la clasificación

del conjunto de datos Seismic-bump aplicado a 5 métodos de clasificación. De acuerdo a los métodos que tuvieron un mayor porcentaje de clasificación de instancias, se puede apreciar que para este caso existen dos métodos con un indice alto de clasificación fueron Árboles de Decisión y Máquinas de Soporte de Vectores. Por otro lado, es importante considerar el tiempo que tardaron en clasificar estos métodos, como se puede apreciar en la Figura 3.12, el método de Árboles de Decisión se tardó menos tiempo que el método de Máquinas de Soporte de Vectores. En los demás métodos de clasificación empleados, se observo que su precisión fue similar, sin embargo su porcentaje de error de clasificación fue mayor a los 2 métodos con mayor porcentaje de clasificación correcta, debido a que tenían errores al momento de clasificar, por otro lado, el método de Naïve Bayes fue el que obtuvo un porcentaje mayor de clasificación incorrecta, ésto se observa en las matrices de que se obtuvieron en la aplicación de los métodos. Por lo tanto, para clasificar este conjunto de datos siempre será mejor utilizar el método de Árboles de Decisión debido a que obtuvo un porcentaje considerable de clasificación y el tiempo que demoro en clasificar es mínimo en comparación al del otro método.

3.7. Gama Mágica Telescopio

Los datos son generados MC (ver más abajo) para simular el registro de partículas de alta energía gamma en un telescopio atmosférica gamma de Cherenkov con base en tierra utilizando la técnica de formación de imágenes. Telescopio gamma de Cherenkov observa rayos gamma de alta energía, tomando ventaja de la radiación emitida por las partículas cargadas producidas en el interior de las duchas electromagnéticas iniciadas por las gammas, y el desarrollo en la atmósfera. Esta radiación Cherenkov (de visible a longitudes de onda UV) se filtra a través de la atmósfera y se graba en el detector , lo que permite la reconstrucción de los parámetros de la ducha . La información disponible se compone de pulsos dejados por los fotones Cherenkov entrantes en los tubos fotomultiplicadores, dispuestos en un plano, la cámara. Dependiendo de la energía de la gamma primaria, un total de unos pocos cientos a unos 10.000 fotones Cherenkov conseguir recoge, en patrones (llamados la imagen ducha) , lo que permite discriminar estadísticamente las causadas por radiaciones gamma primarios (señal) de las imágenes de duchas hadrónicas inició por los rayos cósmicos en la atmósfera superior (fondo). Típicamente, la imagen de una ducha después de algún procesamiento previo es un racimo alargado. Su eje longitudinal está orientado hacia el centro de la cámara si el eje de la ducha es paralelo al eje óptico del telescopio, es decir, si el eje del telescopio se dirige hacia una fuente de punto. Un análisis de componentes principales se lleva a cabo en el plano de la cámara, lo que resulta en un eje de correlación y define una elipse. Si las deposiciones se distribuyeron como gaussiana bi variante, esto sería una elipse equidensity. Los parámetros característicos de esta elipse (a menudo llamadas parámetros Hilla) son algunos de los parámetros de imagen que pueden ser utilizados para la discriminación. Las deposiciones de energía son típicamente asimétrica a lo largo del eje mayor, y esta asimetría también puede ser utilizada en la discriminación. Hay, además, discriminar aún más características, como la medida de la agrupación en el plano de la imagen, o la suma total de las deposiciones[1].

Tabla 3.13: Número de Instancias y Atributos de Gama Mágica Telescopio

Instancias	Atributos
19020	11

Tabla 3.14: Atributos de Gama Mágica Telescopio

Atributo	Tipo	Dominio
Eje ancho	Real	18-50
Eje menor	Real	1-4
Log todos pixeles	Binario	1,0
Suma 2 pixeles	Real	0.2323-0.24313
Pixel mas alto	Real	1,2,3,4
Distancia pixel alto	Real.	0.365, 0.600, 0.666, etc.
longitud mayor	Real	1.2, 0.4678, ...
longitud menor	Real	0.4755, 0.45, 0.33, ...
Angulo	Real	0.4755, 0.4005, 0.3233, ...
Distancia	Real	0.365, 0.444, 0.3200, ...
clases	Real	4,5...,10, 19,... etc.

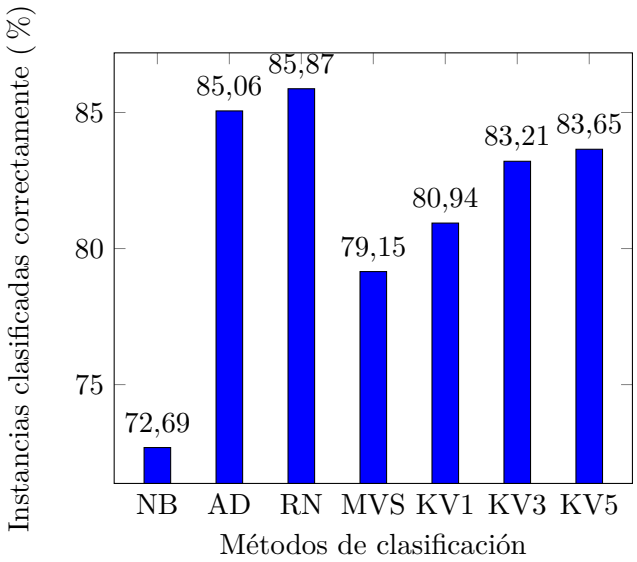

Figura 3.13: Porcentajes de clasificación correcta del conjunto de datos de Gama Mágica Telescopio

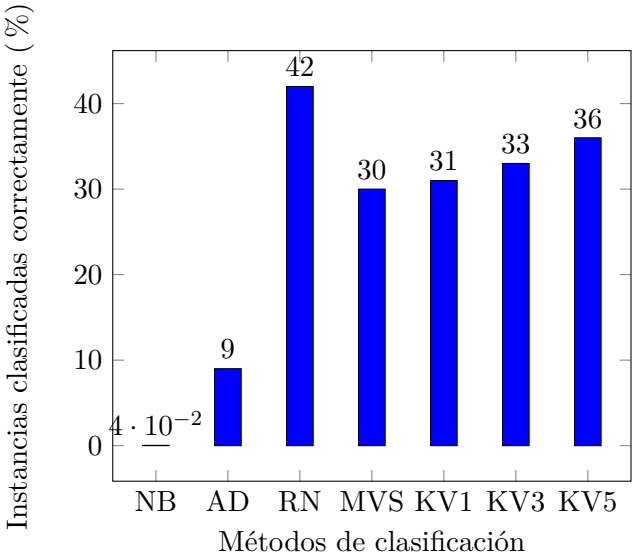

Figura 3.14: Tiempos del procesamiento del conjunto de datos de Gama Mágica Telescopio

Los resultados generales de la clasificación se muestran en la Figura 3.13 donde se puede ver el porcentaje de clasificación correcta que obtiene cada método de clasificación utilizada. la Figura 3.14 muestra por su parte los resultados del tiempo de procesamiento empleado por cada algoritmo mostrado en segundos. Estos resultados son un porcentaje de la clasificación del conjunto de datos Gama Mágica Telescopio aplicado a 5 métodos de clasificación. De acuerdo a los métodos que tuvieron un mayor porcentaje de clasificación de instancias, se puede apreciar que para este caso existen dos métodos con un indice alto de clasificación fueron Árboles de Decisión y Redes Neuronales. Por otro lado, es importante considerar el tiempo que tardaron en clasificar estos métodos, como se puede apreciar en la Figura 3.14, el método de Árboles de Decisión se tardó menos tiempo que el método de Redes Neuronales. En los demás métodos de clasificación empleados, se observo que su precisión fue similar, sin embargo su porcentaje de error de clasificación fue mayor a los 2 métodos con mayor porcentaje de clasificación correcta, debido a que tenían errores al momento de clasificar, por otro lado, el método de Naïve Bayes fue el que obtuvo un porcentaje mayor de clasificación incorrecta, ésto se observa en las matrices de que se obtuvieron en la aplicación de los métodos. Por lo tanto, para clasificar este conjunto de datos siempre será mejor utilizar el método de Árboles de Decisión debido a que obtuvo un porcentaje considerable de clasificación y el tiempo que demoro en clasificar es mínimo en comparación al del otro método.

3.8. Método Anticonceptivo

Este conjunto de datos es un subconjunto de la encuesta de Las muestras son mujeres casadas que eran o no embarazada o no saben si se encontraban en el momento de la entrevista. El problema es predecir la elección actual Método Anticonceptivo (sin uso, métodos de largo plazo, o los métodos de corto plazo) de una mujer sobre la base de sus características demográficas y socio-económicas[1].

Tabla 3.15: Número de Instancias y Atributos de Método Anticonceptivo

Instancias	Atributos
1473	10

Tabla 3.16: Atributos de Método Anticonceptivo

Atributo	Tipo	Dominio
Edad de la esposa	Numérico	18-50
La educación de la esposa	Real	1-4
La educación del esposo	Real	1-4
Número de hijos nacidos vivos	Numérico	1-10
La religión de la esposa	Real	0,1
¿Esposa Ahora está trabajando?	Binario	0,1
Ocupación del esposo	Real	1-4
Esperanza de vida	Real	1-4
Exposición a los medios	Binario	1,0
Clase	Real	1,2,3

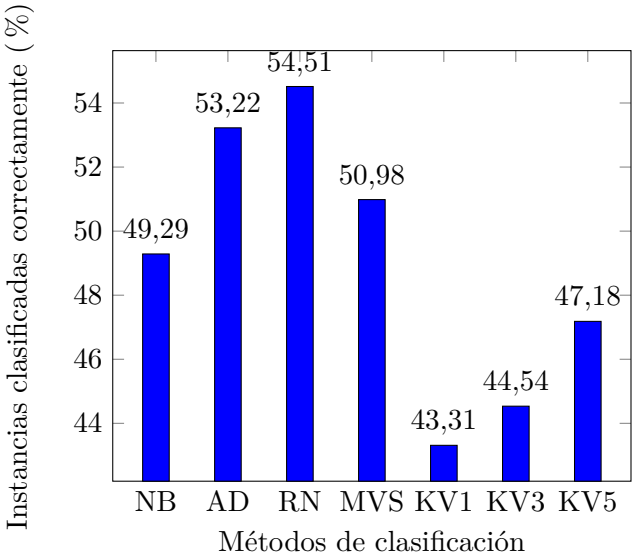

Figura 3.15: Porcentajes de clasificación correcta del conjunto de datos de Método Anticonceptivo

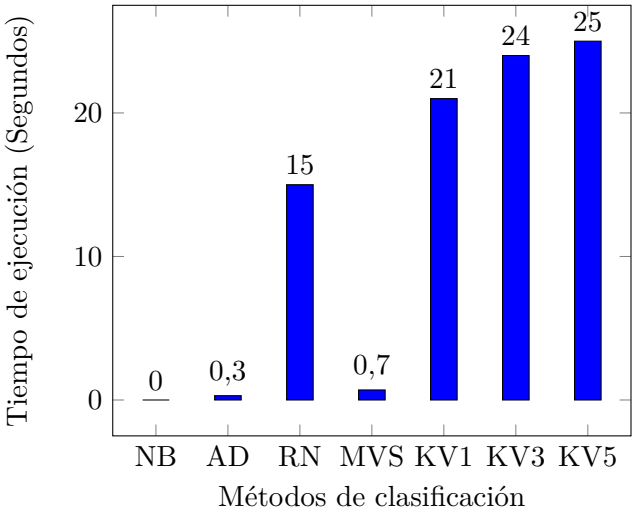

Figura 3.16: Tiempos del procesamiento del conjunto de datos de Método Anticonceptivo

Los resultados generales de la clasificación se muestran en la Figura 3.15 donde se puede ver el porcentaje de clasificación correcta que obtiene cada método de clasificación utilizada. la Figura 3.16 muestra por su parte los resultados del tiempo de procesamiento empleado por cada algoritmo mostrado en segundos. Estos resultados son un porcentaje de la clasifica-

ción del conjunto de datos Método Anticonceptivo aplicado a 5 métodos de clasificación. De acuerdo a los métodos que tuvieron un mayor porcentaje de clasificación de instancias, se puede apreciar que para este caso existen dos métodos con un indice alto de clasificación fueron Árboles de Decisión y Redes Neuronales. Por otro lado, es importante considerar el tiempo que tardaron en clasificar estos métodos, como se puede apreciar en la Figura 3.16, el método de Árboles de Decisión se tardó menos tiempo que el método de Redes Neuronales. En los demás métodos de clasificación empleados, se observo que su precisión fue similar, sin embargo su porcentaje de error de clasificación fue mayor a los 2 métodos con mayor porcentaje de clasificación correcta, debido a que tenían errores al momento de clasificar, por otro lado, el método de K-Vecinos fue el que obtuvo un porcentaje mayor de clasificación incorrecta, ésto se observa en las matrices de que se obtuvieron en la aplicación de los métodos. Por lo tanto, para clasificar este conjunto de datos siempre será mejor utilizar el método de Árboles de Decisión debido a que obtuvo un porcentaje considerable de clasificación y el tiempo que demoro en clasificar es mínimo en comparación al del otro método.

3.9. Letter Recognition

El objetivo de esta base de datos es identificar cada número grande de un rectángulo en blanco y negro como una de las 26 letras capitales en el alfabeto inglés. Los caracteres fueron basados en 20 diferentes fuentes y cada letra dentro de éstas 20 fuentes fue aleatoriamente alterada, distorsionada, para producir un archivo de 20,000 estímulos únicos. Cada estímulo fue convertido en 16 atributos numéricos, los cuales fueron escalados para ajustarlos dentro de un rango de valores enteros (de 0 a 15)[1].

Tabla 3.17: Porcentajes de clasificación correcta del conjunto de datos de Letter Recognition

Instancias	Atributos
20000	17

Tabla 3.18: Atributos de Letter Recognition

Atributo	Tipo	Dominio
Clase	Categórico	A,B,C,D,E,...,X,Y,Z
Poscajax hor	Numérico	9,10,7,3,...
Poscajay ver	Numérico	9,10,7,3,...
Ancho caja	Numérico	9,10,7,3,...
Altura caja	Numérico	9,10,7,3,...
totalpix	Numérico	9,10,7,3,...
Barrax	Numérico	9,10,7,3,...
Barray	Numérico	9,10,7,3,...
Barrax2	Numérico	9,10,7,3,...
Barray2	Numérico	9,10,7,3,...
barraxy	Numérico	9,10,7,3,...
x2ybr	Numérico	9,10,7,3,...
xy2br	Numérico	9,10,7,3,...
x esquina	Numérico	9,10,7,3,...
Xegvy	Numérico	9,10,7,3,...
y esquina	Numérico	9,10,7,3,...
yegvx	Numérico	9,10,7,3,...

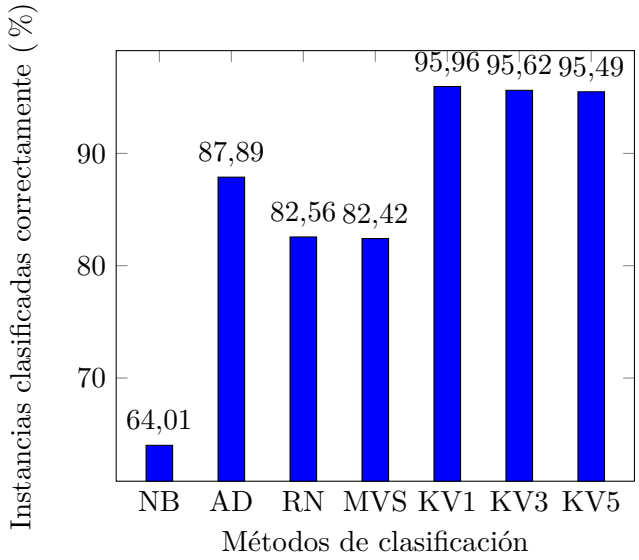

Figura 3.17: Gráfico de resultados del conjunto de datos Letter Recognition

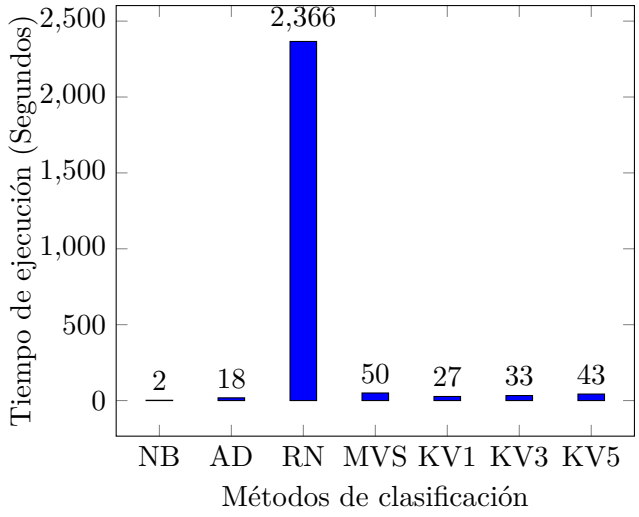

Figura 3.18: Tiempos del procesamiento del conjunto de datos de Letter Recognition

Los resultados generales de la clasificación se muestran en la Figura 3.17 donde se puede ver el porcentaje de clasificación correcta que obtiene cada método de clasificación utilizada. la Figura 3.18 muestra por su parte los resultados del tiempo de procesamiento empleado por cada algoritmo mostrado en segundos. Estos resultados son un porcentaje de la clasificación del conjunto de datos Método Anticonceptivo aplicado a 5 métodos de cla-

sificación. De acuerdo a los métodos que tuvieron un mayor porcentaje de clasificación de instancias, se puede apreciar que para este caso existen dos métodos con un indice alto de clasificación fueron K-Vecinos **1** y K-Vecinos **2**. Por otro lado, es importante considerar el tiempo que tardaron en clasificar estos métodos, como se puede apreciar en la Figura 3.18, el método de K-Vecinos **1** se tardó menos tiempo que el método de K-Vecinos **2**. En los demás métodos de clasificación empleados, se observo que su precisión fue similar, sin embargo su porcentaje de error de clasificación fue mayor a los 2 métodos con mayor porcentaje de clasificación correcta, debido a que tenían errores al momento de clasificar, por otro lado, el método de Naïve Bayes fue el que obtuvo un porcentaje mayor de clasificación incorrecta, ésto se observa en las matrices de que se obtuvieron en la aplicación de los métodos. Por lo tanto, para clasificar este conjunto de datos siempre será mejor utilizar el método de K-Vecinos **1** debido a que obtuvo un porcentaje considerable de clasificación y el tiempo que demoro en clasificar es mínimo en comparación al del otro método.

3.10. Banknote

Los datos fueron extraídos de imágenes que fueron tomadas de los especímenes genuinos y falsificados y como de billetes de banco. Para la digitalización, una cámara fotográfica industrial usualmente destinada para inspección estampada fue usada. Las imágenes finales tienen a 400x 400 pixeles. Debido al lente del objeto y distancia para las fotos del objeto de escalas de grises investigadas con una decisión de aproximadamente 660 puntos por pulgada fue ganado. La herramienta de la ola pequeña Transform se usara para extractar características de imágenes[1].

Tabla 3.19: Número de Instancias y Atributos de Banknote

Instancias	Atributos
1372	5

Tabla 3.20: Atributos de Banknote

Atributo	Tipo	Dominio
varianza imagen	Real	-8,..6
asimetria imagen	Real	-14,..13
curtosis imagen	Real	-6,..18
entropia	Real	-9,..3
clase	Real	0,1

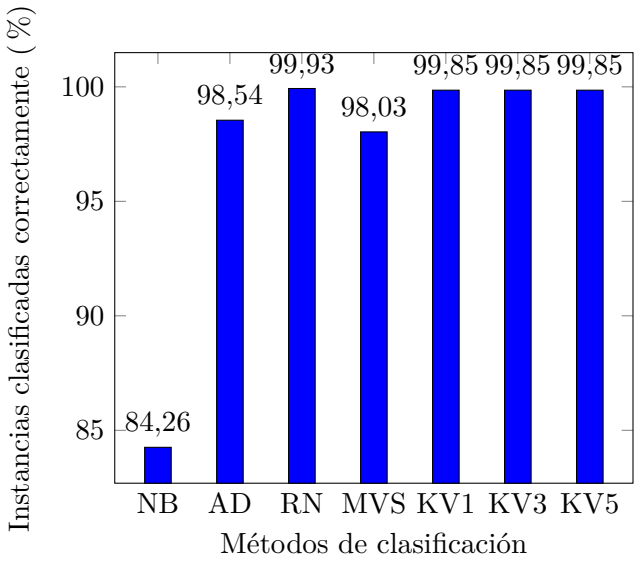

Figura 3.19: Porcentajes de clasificación correcta del conjunto de datos de Banknote

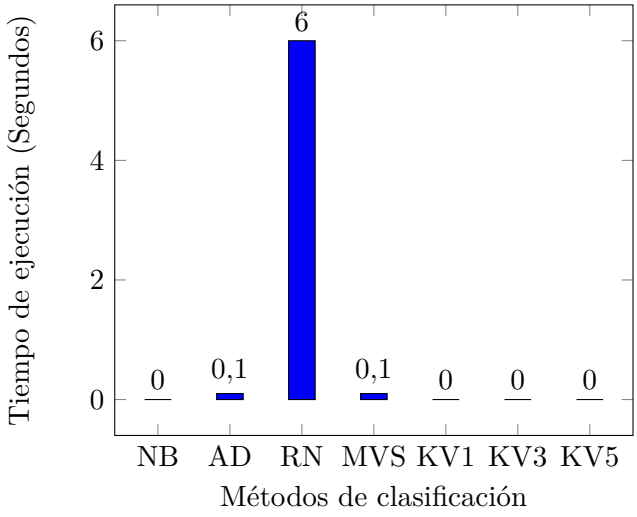

Figura 3.20: Tiempos del procesamiento del conjunto de datos de Banknote

Los resultados generales de la clasificación se muestran en la Figura 3.19 donde se puede ver el porcentaje de clasificación correcta que obtiene cada método de clasificación utilizada. la Figura 3.20 muestra por su parte los resultados del tiempo de procesamiento empleado por cada algoritmo mostrado en segundos. Estos resultados son un porcentaje de la clasificación del conjunto de datos Método Anticonceptivo aplicado a 5 métodos de cla-

sificación. De acuerdo a los métodos que tuvieron un mayor porcentaje de clasificación de instancias, se puede apreciar que para este caso existen dos métodos con un indice alto de clasificación fueron Redes Neuronales y K-Vecinos **1**. Por otro lado, es importante considerar el tiempo que tardaron en clasificar estos métodos, como se puede apreciar en la Figura 3.20, el método de K-Vecinos **1** se tardó menos tiempo que el método de Redes Neuronales. En los demás métodos de clasificación empleados, se observo que su precisión fue similar, sin embargo su porcentaje de error de clasificación fue mayor a los 2 métodos con mayor porcentaje de clasificación correcta, debido a que tenían errores al momento de clasificar, por otro lado, el método de Naïve Bayes fue el que obtuvo un porcentaje mayor de clasificación incorrecta, ésto se observa en las matrices de que se obtuvieron en la aplicación de los métodos. Por lo tanto, para clasificar este conjunto de datos siempre será mejor utilizar el método de K-Vecinos **1** debido a que obtuvo un porcentaje considerable de clasificación y el tiempo que demoro en clasificar es mínimo en comparación al del otro método.

Conclusiones

La minería de datos se aplica principalmente en las áreas financieras, de mercado, seguro, educación, medicina y biología, algunas compañías han optado por aplicar dicha técnica en sus almacenes de datos, esto con la finalidad de obtener mayor eficiencia y eficacia en cuanto a las decisiones tomadas.

El proyecto realizado ha generado una serie de resultados los cuales contribuyen a un nuevo aprendizaje, se aplicaron 5 métodos de clasificación supervisada: Naïve Bayes, Árboles de Decisión, Redes Neuronales, K-Vecinos con 1, 3, y 5 vecinos y Maquina de Vectores de Soporte, los cuales ayudaron para el análisis de los conjuntos de datos.

La forma de evaluación utilizada para los 5 métodos, fue: Cross Validation (validación cruzada) con 10 folds, Naïve Bayes fue uno de los métodos que obtuvo menor porcentaje de clasificación correcta, por otro lado, uno de los métodos que más tardo fue Redes Neuronales, sin embargo, obtuvo mejores resultados de clasificación correcta en comparación a otros.

Los métodos Árboles de decisión y Maquinas de Soporte de Vectores fueron los que obtuvieron mayor porcentaje de clasificación correcta, pero Maquinas de Soporte de Vectores tardo más tiempo en clasificar.

Finalmente se concluye que el método más óptimo para estos conjuntos de datos siempre será Árboles de Decisión tomando como fundamento el resultado que arrojo; tanto en clasificación correcta, como en el tiempo que tardo en clasificar.

Bibliografía

[1] Sitio web Donald Bren School of Information and Computer Sciences @ UC Irvine: http://archive.ics.uci.edu/ml/datasets.html/ , 11 de marzo de 2014

[2] Tutorial Weka: http://isa.umh.es/asignaturas/crss/turorialWEKA.pdf , 30 de Abril de 2014

[3] Documentación de Weka: http://www.cs.waikato.ac.nz/~ml/weka/documentation.html , 29 de Abril de 2014

[4] Han, J.,*Data Mining Concepts and Techniques*. 500 Sansome Street, Suite 400, San Francisco, CA 94111, 2nd edition, 2006.

[5] *Sitio web Waikaton, Attribute-Relation File Format (ARFF*http://www.cs.waikato.ac.nz/ml/weka/arff.html, 5 de Mayo de 2014

[6] *Sitio web Uvigo, uso de WEKA Evaluación de técnicas de combinación de clasificadores*http://ccia.ei.uvigo.es/docencia/MRA/practicas/practica-1/practica-1.html , 6 de mayo de 2014.