

Evaluation of Classification Algorithms using Cross Validation

Leticia Laura-Ochoa, Mg¹

¹Universidad Nacional de San Agustín de Arequipa, Perú, llaurao@unsa.edu.pe

Abstract -- Cross validation allows to evaluate the accuracy of the classification algorithms with low error, but has the problem of computational cost versus large volumes of data. In this work, the serial and parallel implementation of leave-one-out and k-fold cross validation techniques is performed using the R software environment. A comparison between the precision and error results obtained with cross validation techniques is presented, as well as the execution time thereof, reducing in the parallel implementation.

Keywords – classification algorithms, cross validation, parallel computing.

Digital Object Identifier (DOI):
<http://dx.doi.org/10.18687/LACCEI2019.1.1.471>
ISBN: 978-0-9993443-6-1 ISSN: 2414-6390

Evaluación de Algoritmos de Clasificación utilizando Validación Cruzada

Leticia Laura-Ochoa, Mg¹

¹Universidad Nacional de San Agustín de Arequipa, Perú, llaurao@unsa.edu.pe

Resumen—La validación cruzada permite evaluar la precisión de los algoritmos de clasificación con bajo error, pero tiene el problema de alto costo computacional para grandes volúmenes de datos. En este trabajo se realiza la implementación serial y paralela de las técnicas de validación cruzada dejando uno afuera y de K iteraciones utilizando el entorno de software R. Se presenta una comparación entre los resultados de precisión y error obtenidos a partir de las técnicas de validación cruzada, así como el tiempo de ejecución de los mismos, reduciéndose en la implementación paralela.

Palabras claves—algoritmos de clasificación, validación cruzada, computación paralela

I. INTRODUCCIÓN

La clasificación es una técnica esencial que se utiliza para predecir los valores de clase de las nuevas instancias, observaciones o registros [1], la cual tiene varias aplicaciones, entre las que se incluyen detección de fraudes, predicción del rendimiento, fabricación, diagnóstico médico [2], es muy utilizada en minería de datos y machine learning.

Además, los modelos de clasificación cumplen dos funciones importantes en la minería de datos: Como modelo predictivo para clasificar instancias previamente sin etiquetar, y como modelo descriptivo para identificar las características que distinguen instancias de diferentes clases [3].

Por otro lado, la selección de modelos es un paso indispensable en el proceso de desarrollar un modelo de predicción funcional o un modelo para comprender el mecanismo de generación de datos [4].

Un buen modelo de clasificación debe proporcionar predicciones precisas con un tiempo de respuesta rápido [3]. En este sentido, la precisión de un clasificador en un conjunto de pruebas dado es el porcentaje de tuplas del conjunto de pruebas que están clasificadas correctamente por el clasificador [2]. Además, al evaluar algoritmos de clasificación, hay diferentes formas de medir su rendimiento [5], la matriz de confusión es una matriz n por n , donde n es el número de clases, la cual permite analizar qué tan bien un algoritmo de clasificación puede reconocer instancias de diferentes clases, tiene información sobre las clasificaciones reales y predicciones de clases hechas por el modelo de clasificación [2][6]

Los resultados de precisión y error de los modelos predictivos en minería de datos que se calculan a partir de la matriz de confusión pueden variar, debido a que las muestras

validación cruzada para comparar los resultados de los algoritmos de clasificación y seleccionar presumiblemente de forma más certera el modelo que realice la predicción con mayor precisión [7]

Se debe tener en cuenta que la validación cruzada no produce ningún clasificador final único, y la matriz de confusión que proporciona no es el rendimiento de ningún clasificador único específico. En cambio, esta matriz es una estimación del rendimiento promedio de un clasificador [5]. Las técnicas de validación cruzada dejando uno afuera o k iteraciones son muy populares para evaluar el rendimiento de los algoritmos de clasificación [1].

Con la validación cruzada el error es muy bajo, pero en cambio, a nivel computacional es muy costoso [5], puesto que a mayor volumen de datos, el tiempo de ejecución se incrementa y se realiza un elevado número de iteraciones, razón por la cual es conveniente utilizar la programación paralela para disminuir el tiempo de procesamiento.

Según [8], Python y R son dos de los lenguajes más utilizados para el análisis de datos y su procesamiento mediante técnicas de minería de datos. En este trabajo se utiliza R, el cual es un entorno de software libre para cálculos estadísticos y gráficos, puede ser utilizado en una amplia variedad de plataformas como UNIX, Windows y MacOS, proporciona varias técnicas estadísticas como modelado lineal y no lineal, análisis de series de tiempo, pruebas estadísticas clásicas, clasificación, agrupación en clústeres, etc. [9][10]. De otro lado, se utiliza el paquete snow ("Simple Network of Workstations"), el cual es probablemente el paquete de programación paralelo más popular disponible para R [11], se enfoca en un modelo de repartir-calcular-recolectar en el cual un proceso maestro divide una tarea en subtareas que se asignan a una colección de procesos de trabajadores, que luego ejecutan las tareas y devuelven sus resultados al proceso maestro [12].

Este trabajo está organizado de la siguiente manera. En la sección II se presenta marco conceptual relacionado a validación cruzada. La sección III presenta la descripción de la implementación de la técnica de validación cruzada en R tanto serial como paralelo. La sección IV Resultados, proporciona una comparación con respecto al tiempo de ejecución, precisión y error de las técnicas de validación cruzada. Finalmente, se expone las conclusiones y apreciaciones finales del trabajo realizado.

Digital Object Identifier (DOI):

<http://dx.doi.org/10.18687/LACCEI2019.1.1.471>

ISBN: 978-0-9993443-6-1 ISSN: 2414-6390

II. MARCO CONCEPTUAL

A. Validación cruzada

Cuando la cantidad de datos para entrenamiento y prueba es limitada, se puede aplicar el método de retención (holdout) para estimar el error, el cual reserva un porcentaje de datos para la prueba y el resto para entrenamiento, se suele considerar un tercio de datos para la prueba y los otros dos tercios restantes para entrenamiento [13]. Con este método puede existir una variación en la estimación del error y no resultar representativos los datos de entrenamiento y prueba, los cuales suelen ser generados al azar, una mejora del método de retención es la utilización de la validación cruzada.

La validación cruzada es un método de evaluación de modelos ampliamente utilizado que tiene como objetivo hacer un uso efectivo de todas las instancias etiquetadas en D tanto para entrenamiento como para pruebas [3], permite utilizar todos los registros disponibles para entrenamiento y al mismo tiempo usar muchos registros como conjuntos de pruebas independientes [5].

Además, con la validación cruzada, se decide el número fijo de particiones de los datos. Supongamos que se elige tres. Luego, los datos se dividen en tres particiones aproximadamente iguales y cada una a su vez se usa para pruebas y el resto se usa para entrenamiento. Es decir, utilice dos tercios para el entrenamiento y un tercio para las pruebas y repita el procedimiento tres veces para que, al final, cada instancia se haya utilizado exactamente una vez para las pruebas. Esto se denomina validación cruzada triple [13].

B. Validación cruzada dejando uno afuera

El resultado de la validación cruzada dejando uno fuera o Leave-one-out cross-validation (LOOCV) es una matriz de confusión basada en el uso de cada registro etiquetado como registro de prueba una sola vez. Cuando se usa un registro para probar el clasificador, este no se ha utilizado para entrenar a ese clasificador. Por lo tanto, la matriz de confusión obtenida por la validación cruzada es intuitivamente un indicador justo del rendimiento del algoritmo de aprendizaje en registros de pruebas independientes, el error es muy bajo; sin embargo, la complejidad del tiempo de la validación cruzada es k veces la de ejecutar el algoritmo de entrenamiento una vez, por lo que a menudo la LOOCV es computacionalmente costoso e inviable, ya que se realiza N iteraciones, donde N es el número de instancias que se tengan. En una investigación reciente, la opción más común para k es 10 [5].

En la LOOCV cada instancia, a su vez, queda fuera, y el método de aprendizaje es entrenado sobre todas las instancias restantes. Se juzga por su corrección en la instancia restante: uno o cero para el éxito o el fracaso, respectivamente. Esto se repite para las n instancias, luego se promedian los resultados y ese promedio representa la estimación de error final [13].

Este procedimiento permite utilizar la mayor cantidad de registros para el entrenamiento, lo que presumiblemente aumenta la probabilidad de que el clasificador sea exacto, no

se utiliza muestras al azar, se obtiene el mismo resultado; sin embargo tiene alto costo computacional, por lo que no es factible para grandes conjuntos de datos. Se puede obtener una estimación lo más precisa posible para pequeños conjuntos de datos [13].

C. Validación cruzada de k iteraciones

La validación cruzada de k iteraciones o k -fold cross validation consiste en dividir los datos originales en k subconjuntos y, al momento de realizar el entrenamiento, se va a tomar cada k subconjunto como conjunto de prueba del modelo, mientras que el resto de subconjuntos ($k-1$) se tomará como conjunto de entrenamiento. El proceso de validación cruzada se repetirá k veces, y en cada iteración se tomará un conjunto de prueba diferente, siendo el resto de datos el conjunto de entrenamiento. Al finalizar las k iteraciones, se calcula el promedio de los resultados de precisión y error obtenidos para cada subconjunto de prueba [14].

En la validación cruzada k -fold, los datos iniciales se dividen aleatoriamente en k subconjuntos mutuamente exclusivos, D_1, D_2, \dots, D_k , cada uno aproximadamente del mismo tamaño. El entrenamiento y las pruebas se realizan k veces [13]. En general, se recomienda una validación cruzada estratificada de 10-fold para estimar la precisión, debido a su sesgo y varianza relativamente bajos. [2].

III. DESCRIPCIÓN DEL TRABAJO

En este trabajo se realizó la evaluación de algoritmos de clasificación de minería de datos utilizando las técnicas de validación cruzada dejando uno afuera, llamado también Leave-one-out cross-validation (LOOCV) y validación cruzada de K iteraciones, llamada también K -fold cross-validation. Se realizaron para ambos casos pruebas de ejecución serial y en paralelo. La implementación de la validación cruzada se realizó utilizando el lenguaje de programación R.

El archivo de notas de extensión .CSV con el que se realizaron las pruebas contiene como campos: El código del alumno, nombres de las asignaturas y rendimiento, este último campo se utilizó como variable objetivo o variable dependiente para la clasificación del rendimiento académico.

Las técnicas de clasificación que se utilizaron para las pruebas de validación cruzada son: Árbol de Decisión, Redes Bayesianas, K Nearest Neighbors (KNN) – K vecinos más cercanos y Support Vector Machine (SVM) - Máquinas de Soporte Vectorial.

A. Validación cruzada dejando uno afuera

En el trabajo [7], se encuentra la implementación serial de la validación cruzada dejando uno afuera.

En cada iteración de LOOCV se considera solamente un registro para la prueba y el resto de registros para el entrenamiento.

Se utilizó un conjunto de datos de 69 registros para las pruebas de validación cruzada dejando uno afuera, por lo que hubieron 69 iteraciones.

En cada iteración se obtuvieron cuatro resultados de precisión y error correspondientes a las cuatro técnicas de minería de datos aplicadas. Luego se calcularon la media aritmética de los resultados por cada técnica de clasificación. Se iteró cinco veces para verificar que el error es bajo, resultando para todos los algoritmos un valor constante, no se iteró más veces debido a que se obtienen los mismos resultados.

En la Tabla 1 se muestra los resultados de precisión obtenidos a partir de la técnica de validación cruzada dejando uno afuera (LOOCV).

TABLA 1
PRECISIÓN DE LOS MODELOS - LOOCV

Modelo	Iteración 1	Iteración 2	Iteración 3	Iteración 4	Iteración 5
Árbol de Decisión	86.95652	86.95652	86.95652	86.95652	86.95652
Redes Bayesianas	97.10145	97.10145	97.10145	97.10145	97.10145
KNN	95.65217	95.65217	95.65217	95.65217	95.65217
SVM	100.0000	100.0000	100.0000	100.0000	100.0000

En la Fig. 1 se observa que con la técnica de clasificación SVM, se obtuvo mayor precisión, obteniendo un 100% de aciertos en las clasificaciones.

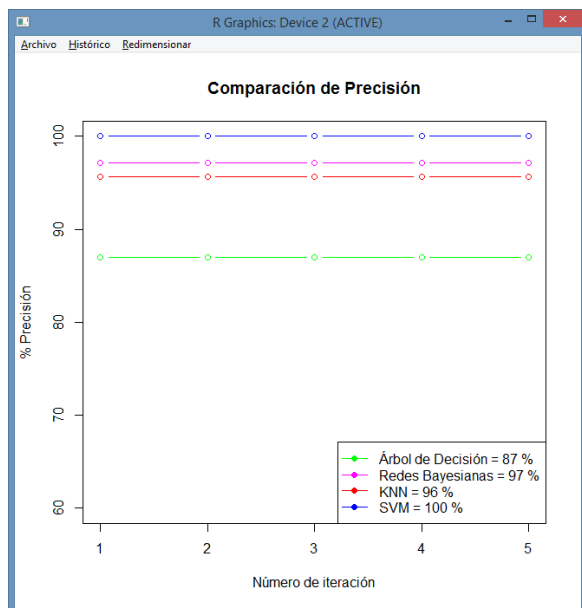


Fig. 1 Precisión de los modelos - LOOCV

En la Tabla 2 se muestra los resultados de error obtenidos a partir de la técnica de validación cruzada dejando uno afuera (LOOCV).

TABLA 2
ERROR DE LOS MODELOS - LOOCV

Modelo	Iteración 1	Iteración 2	Iteración 3	Iteración 4	Iteración 5
Árbol de Decisión	13.04348	13.04348	13.04348	13.04348	13.04348
Redes Bayesianas	2.898551	2.898551	2.898551	2.898551	2.898551
KNN	4.347826	4.347826	4.347826	4.347826	4.347826
SVM	0.000000	0.000000	0.000000	0.000000	0.000000

En la Fig. 2 se observa que con la técnica de clasificación SVM, se obtuvo menor error.

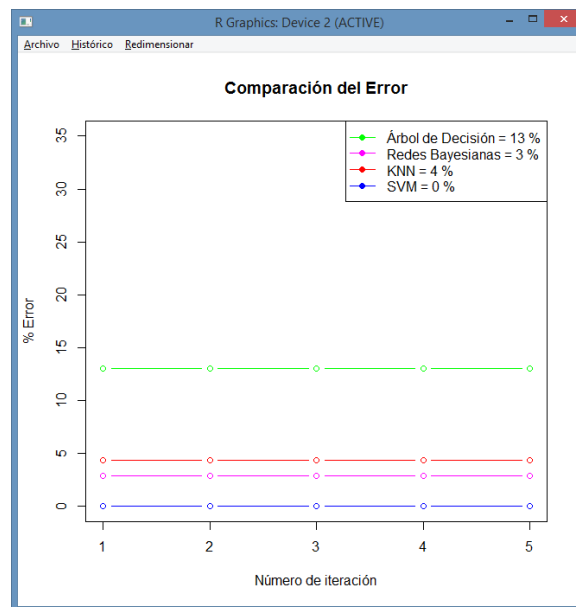


Fig. 2 Error de los modelos - LOOCV

En la Tabla 3 se muestra la precisión y error promedio obtenido a partir de la validación cruzada dejando uno afuera para las técnicas de clasificación de minería de datos utilizados.

TABLA 3
PRECISIÓN Y ERROR PROMEDIO DE LOS MODELOS - LOOCV

Modelo	% Precisión	% Error
Árbol de Decisión	87%	13%
Redes Bayesianas	97%	3%
KNN	96%	4%
SVM	100%	0%

Con esta técnica de validación cruzada dejando uno afuera, el error es muy bajo; pero tiene un alto costo computacional por la cantidad de iteraciones que se pueden requerir dependiendo del número total de registros. Es por esta razón que se investigó el uso de computación paralela para reducir el tiempo de ejecución.

Se utilizó el paquete snow (“Simple Network of Workstations”), el cual permite la programación paralela en R.

El paquete snow proporciona soporte para ejecutar fácilmente las funciones R en paralelo[8]. Se creó un cluster de cuatro trabajadores utilizando el transporte socket. En la Fig. 3 se muestra un fragmento de la implementación paralela, en la cual la función clusterApply toma como primer argumento el cluster creado, como segundo argumento los diferentes algoritmos de clasificación a ser distribuidos para su ejecución entre los cuatro procesadores.

```
results <- clusterApply(cl,algoritmos, function(param_model) {
  lstResult <- calcularPrecision(param_model, notas, f)
  precision <- lstResult[1]
  error <- lstResult[2]
  lista <- list(tipo <- param_model, precision <- precision, error <- error)
})
```

Fig. 3 Función clusterApply para validación cruzada dejando uno afuera.

Se obtuvieron los mismos resultados de precisión y error que la técnica de validación cruzada dejando uno afuera ejecutada de forma serial.

En la Tabla 4 se muestra el tiempo de ejecución tomado por la técnica de validación cruzada dejando uno afuera para la evaluación de algoritmos de clasificación de forma serial y paralela. Para hallar el tiempo de ejecución se utilizó la función proc.time utilizando el lenguaje R.

TABLA 4
COMPARACIÓN DE TIEMPO DE EJECUCIÓN - LOOCV

Técnica	User time	System time	Elapsed time
LOOCV Serial	20.89	0.01	21.09
LOOCV Paralelo	0.88	0.35	15.83

Donde:

- User time, es el tiempo que demora la CPU en la ejecución de las instrucciones para la generación del modelo.
- System time, es el tiempo tomado por el sistema operativo.
- Elapsed time, es el tiempo total desde que se inició el proceso.

B. Validación cruzada de K iteraciones

Conocido también como K-fold cross-validation, en la cual los datos de muestra se dividen en K subconjuntos. En cada iteración uno de los subconjuntos se utiliza para la tabla de prueba y el resto (K-1) para la tabla de entrenamiento.

Se aplicó esta técnica de validación cruzada de K iteraciones para comparar los resultados de precisión y error de las técnicas de clasificación, así como tiempos de ejecución. Se utilizaron cinco grupos, por lo que un grupo del total de datos se utilizó para la tabla de pruebas y los cuatro grupos restantes para la tabla de entrenamiento.

En cada iteración se obtuvieron cuatro resultados de precisión y error correspondientes a las cuatro técnicas de

minería de datos aplicadas. Luego se calcularon la media aritmética de los resultados por cada técnica de clasificación. Se itero cinco veces para verificar la variabilidad de los valores, resultando un error bajo.

Los resultados de precisión obtenidos con la técnica de validación cruzada de K iteraciones (K-fold) se muestran en la Tabla 5:

TABLA 5
PRECISIÓN DE LOS MODELOS – K-FOLD

Modelo	Iteración 1	Iteración 2	Iteración 3	Iteración 4	Iteración 5
Árbol de Decisión	78.71795	87.28571	88.21245	90.00916	87.09890
Redes Bayesianas	95.83333	91.79487	97.03297	95.71429	92.71795
KNN	92.79762	95.90476	94.27106	93.03571	91.17949
SVM	100.00000	97.00000	100.00000	100.00000	98.66667

En la Fig. 4 se observa que se obtuvo mayor precisión aplicando la técnica de clasificación SVM, obteniendo un 99% de aciertos en las clasificaciones.

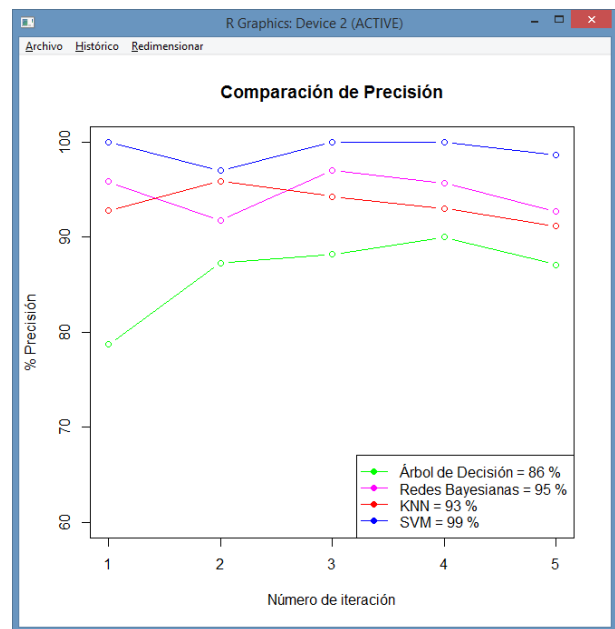


Fig. 4 Precisión de los modelos – K-fold

Los resultados de error obtenidos con la técnica de validación cruzada de K iteraciones (K-fold) se muestran en la Tabla 6:

TABLA 6
ERROR DE LOS MODELOS – K-FOLD

Modelo	Iteración 1	Iteración 2	Iteración 3	Iteración 4	Iteración 5
Árbol de Decisión	21.282051	12.714286	11.787546	9.990842	12.901099
Redes Bayesianas	4.166667	8.205128	2.967033	4.285714	7.282051
KNN	7.202381	4.095238	5.728938	6.964286	8.820513
SVM	0.000000	3.000000	0.000000	0.000000	1.333333

En la Fig. 5 se observa que se obtuvo menor error aplicando la técnica de clasificación SVM, obteniendo 0.87% de error en las predicciones.

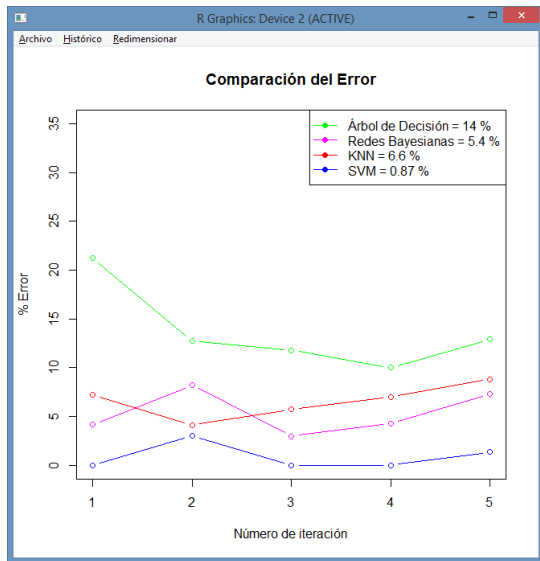


Fig. 5 Error de los modelos – K-fold

En la Tabla 7 se muestra un resumen de los porcentajes promedios de precisión y error obtenidos con la aplicación de las técnicas de aprendizaje supervisado de minería de datos seleccionados.

TABLA 7
PRECISIÓN Y ERROR PROMEDIO DE LOS MODELOS – K-FOLD

Modelo	% Precisión	% Error
Árbol de Decisión	86%	14%
Redes Bayesianas	95%	5%
KNN	93%	7%
SVM	99%	1%

Esta técnica de validación cruzada es confiable ya que el error es muy bajo. Es menos certero que la técnica de validación cruzada dejando uno afuera; pero no varían mucho los valores obtenidos y computacionalmente es menos costoso.

Se utilizó el paquete snow (“Simple Network of Workstations”), el cual permite la programación paralela en R.

De forma similar a la validación cruzada dejando uno afuera, se creó un cluster de cuatro trabajadores utilizando el transporte socket. En la Fig. 6 se muestra un fragmento de la implementación paralela, la variación con respecto al anterior (Fig. 3) se encuentra en el tercer argumento de la función calcularPrecisión, en este caso se envía una muestra formado por uno de los grupos de K-fold, y no el registro individual para la tabla prueba.

```

results <- clusterApply(cl,algoritmos, function(param_model) {
  1stResult <- calcularPrecisión(param_model, notas, muestra)
  precision <- 1stResult[1]
  error <- 1stResult[2]
  lista <- list(tipo <- param_model, precision <- precision, error <- error)
})

```

Fig. 6 Función clusterApply para validación cruzada K-fold.

Se obtuvieron resultados similares de precisión y error que la técnica de validación cruzada de K iteraciones ejecutada de forma serial.

En la Tabla 8 se muestra el tiempo de ejecución tomado por la técnica de validación cruzada de K iteraciones para la evaluación de algoritmos de clasificación de forma serial y paralela. Para hallar el tiempo de ejecución se utilizó la función proc.time utilizando el lenguaje R.

TABLA 8
COMPARACIÓN DE TIEMPO DE EJECUCIÓN – K-FOLD

Técnica	User time	System time	Elapsed time
K-fold Serial	1.57	0.00	4.36
K-fold Paralelo	0.14	0.07	1.29

IV. RESULTADOS

En la Tabla 9 se muestra el tiempo de ejecución que demoraron las técnicas de validación cruzada vistas en la sección anterior, donde se observa que hay una reducción en las implementaciones paralelas.

TABLA 9
COMPARACIÓN DE TIEMPO DE EJECUCIÓN

Técnica	User time	System time	Elapsed time
LOOCV Serial	20.89	0.01	21.09
LOOCV Paralelo	0.88	0.35	15.83
K-fold Serial	1.57	0.00	4.36
K-fold Paralelo	0.14	0.07	1.29

Finalmente, se hizo una comparación de la precisión y error de las técnicas de validación cruzada en la evaluación de rendimiento del algoritmo de clasificación de máquinas de soporte vectorial. En la Tabla 10 se muestra los resultados, donde se observa que la técnica de validación cruzada dejando uno afuera (LOOCV) obtiene mejor precisión; sin embargo el resultado de la validación cruzada de k iteraciones también proporciona una buena precisión.

TABLA 10
PRECISIÓN Y ERROR PROMEDIO DE VALIDACIÓN CRUZADA

Técnica	% Precisión	% Error
K-fold	96%	4%
LOOCV	100%	0%

En la Fig. 7 se observa que se obtuvo mayor precisión utilizando la técnica de validación cruzada dejando uno afuera, obteniendo un 100% de aciertos en las clasificaciones.

V. CONCLUSIONES

Con la técnica de validación cruzada se obtiene bajo error en la evaluación del rendimiento de los algoritmos de clasificación; pero debido a su costo computacional es recomendable su utilización mediante implementación paralela.

Cuando se tenga grandes volúmenes de datos se puede utilizar la validación cruzada de k iteraciones, para obtener clasificaciones de bajo error y con menos costo computacional.

Como trabajo futuro se recomienda investigar el uso de otros paquetes existentes de programación paralela en los entornos de software que son mayormente utilizados para minería de datos.

REFERENCIAS

- [1] T. T. Wong, "Performance evaluation of classification algorithms by k-fold and leave-one-out cross validation," *Pattern Recognition*, vol. 48, no 9, pp. 2839-2846, 2015
- [2] J. Han, M. Kamber and J. Pei, *Data Mining: Concepts and Techniques, 3rd ed.*, San Francisco, CA, EEUU: Morgan Kaufmann, 2011.
- [3] P. Tan, M. Steinbach and V. Kumar, *Introduction to Data Mining*, New York, EEUU: Pearson Education, 2006.
- [4] Y. Zhang and Y. Yang, "Cross-validation for selecting a model selection procedure", *Journal of Econometrics*, vol. 187, no 1, pp. 95-112, 2015.
- [5] C. Elkan, Evaluating classifiers, *San Diego: University of California*, 2012.
- [6] M. Sánchez, Matriz de Confusión, <http://myslide.es/documents/matriz-de-confusion-listo.html>, Revisado en Marzo del 2019.
- [7] L. Laura, K. Rosas and C. Baluarte, "Evaluación de Técnicas de Minería de Datos para la Predicción del Rendimiento Académico," 15th LACCEI International Multi-Conference for Engineering, Education, and Technology, Boca Raton - Florida, USA, 2017.
- [8] Revista digital INESEM, "Python vs R para el análisis de datos", <https://revistadigital.inesem.es/informatica-y-tics/python-r-analisis-datos/>, Revisado en Febrero del 2019.
- [9] The R Project for Statistical Computing, <https://www.r-project.org/>, Revisado en Febrero del 2019.
- [10] F. García, "Aplicación de técnicas de Minería de Datos a datos obtenidos por el Centro Andaluz de Medio Ambiente (CEAMA)," Trabajo Fin de Máster Universitario en Estadística Aplicada, Universidad de Granada, España, 2013.
- [11] E. McCallum and S. Weston, *Parallel R*, "O'Reilly Media, Inc.", 2011.
- [12] L. Tierney, A.J. Rossini and N. Li, "Snow: A parallel computing framework for the R system", *International Journal of Parallel Programming*, vol. 37, no 1, pp. 78-90, 2009.
- [13] I. Witten and E. Frank, *Data Mining: Practical Machine Learning Tools and Techniques, second edition*, San Francisco, CA, EEUU: Morgan Kaufmann, 2005.
- [14] Introducción a la Validación Cruzada (k-fold Cross Validation) en R, http://rstudio-pubs-static.s3.amazonaws.com/405322_6d94d05e54b24ba99438f49a6f8662a9.html, Revisado en Marzo del 2019.

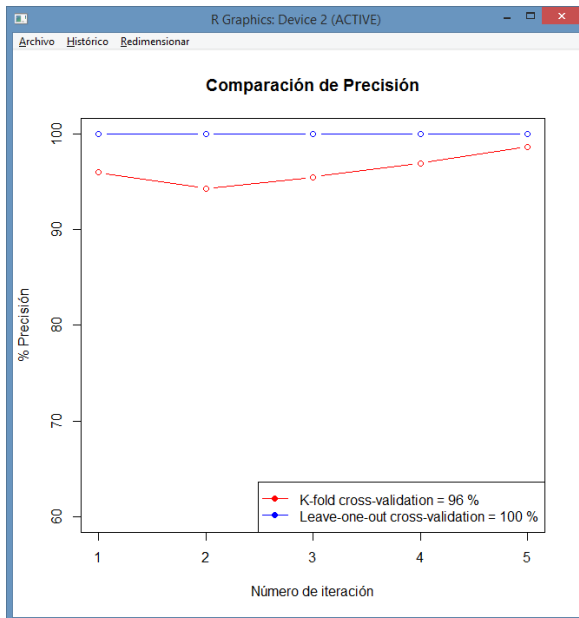


Fig. 7 Comparación de precisión de validación cruzada

En la Fig. 8 se muestra la comparación de error de las técnicas de validación cruzada, no variando demasiado en los resultados, teniendo solamente el 3.7% de diferencia.

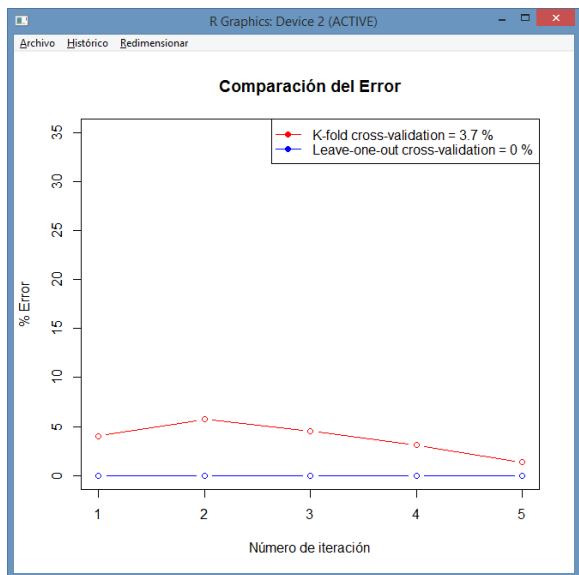


Fig. 8 Comparación de error de validación cruzada

Ambas técnicas de validación cruzada ofrecen buena precisión para la selección del mejor algoritmo de clasificación.

La computación paralela permite disminuir el tiempo de respuesta al utilizar técnicas de validación cruzada que por lo general tienen alto costo computacional.