Predicción de incumplimiento de pago de clientes de tarjetas de crédito, con aplicación del algoritmo del k-vecino más cercano y Clas- FriedmanAligned-ST.

Luis Niebles-Mamani[1], Rodrigo Velarde-Herencia [1] y José Sulla-Torres, Magister. [1,2]
[1]Escuela Profesional de Ingeniería de Sistemas, Universidad Católica de Santa María, Arequipa, Perú,
[2]Universidad Nacional de San Agustín, Arequipa, Perú,
lnieblesm@outlook.com, rodri327@hotmail.com, josullato@gmail.com

Resumen- Las compañías que otorgan tarjetas de crédito a clientes se enfrentan a algunos problemas como es la falta de pago, es por eso que las compañías necesitan controlar tales deudas, de manera que se minimice el riesgo de recuperación de la inversión, a consecuencia de clientes deudores. En este artículo se usó el algoritmo de aprendizaje perezoso KNN, con el método de evaluación estadístico Clas- FriedmanAligned-ST, que nos ayude a predecir el grado de incumplimiento de pago, con el objetivo de optimizar, y mejorar la predicción realizada por algoritmos de minería de datos. La base de datos utilizada para este trabajo contiene 30000 registros, cada uno definido por 25 atributos, de tal cantidad se tomó una muestra significativa de 5439 instancias, con 24 campos. Se desarrolla un modelo de procesamiento de datos, se hace la discusión de los resultados obtenidos; y se concluye con los beneficios de aplicación de computación evolutiva.

Palabras clave- Calificación crediticia, aprendizaje perezoso, algoritmos evolutivos.

Abstract- Companies that give credit cards to clients face some problems such as non-payment, which is why companies need to control such debts, so as to minimize the risk of recovery of the investment, as a result of debtor clients. In this article, the lazy learning algorithm KNN with the method of statistical evaluation Clas- FriedmanAligned-ST was used, to help us to predict the degree of nonpayment of debts, in order to optimize and improve the prediction performed by data mining algorithms. The database used for this work contains 30000 records, each defined by 25 attributes, of which a significant sample of 5439 instances was taken, with 24 fields. A data processing model is developed, the results are discussed; And concludes with the benefits of evolutionary computing application.

Keywords- Credit rating, lazy learning, evolutionary algorithms.

I. Introducción

Las predicciones sobre el incumplimiento de pago de crédito por parte de los clientes son una parte importante de la gestión del riesgo de crédito. Hasta la fecha se han desarrollado e implementado una serie de enfoques al respecto, pero la investigación continúa mejorando las técnicas existentes para obtener soluciones óptimas y diseñar nuevos modelos [1]. Por ello que, a lo largo de los últimos años, las instituciones financieras manejan sistemas de calificación de créditos para la concesión de préstamos, además que el análisis se realiza en base al comportamiento transaccional de esos clientes.

Muchos métodos estadísticos, incluyendo el análisis discriminante, la regresión logística, el clasificador de Bayes y el k-vecino más cercano (KNN), se han utilizado para desarrollar modelos de predicción de riesgo de pago [2]. Con la evolución de la inteligencia artificial y el aprendizaje automático, las redes neuronales artificiales y los árboles de clasificación también se emplearon para predecir el riesgo de crédito [3].

La computación evolutiva según [4] se define como un subconjunto de algoritmos de la inteligencia artificial, con el objetivo de solucionar problemas de optimización, en base a nociones de la selección natural de los seres vivos, propuestas por Charles Darwin. Este nivel de computación, se compone de cuatro campos: algoritmos genéticos, programación evolutiva, estrategias evolutivas, y programación genética. Básicamente, el campo de los algoritmos genéticos consiste en la simulación de procesos genéticos naturales [5], los cuales son: selección (solución factible a un problema dado), cruce (los descendientes de los individuos del proceso, heredan atributos de sus antecesores), y mutación (cambio aleatorio de un gen que implica posibles mejoras). Por el lado de la programación evolutiva, esta es utilizada para la resolución de problemas de predicción, con bases de máquinas de estados finitos, todo ello para el desarrollo de modelos de comportamiento. Las estrategias de evolución están fundadas en la metaevolución del aspecto fenotípico evolutivo; estas trabajan de manera muy similar a lo realizado por los algoritmos genéticos, con la diferencia de operar con una mayor cantidad de objetos descendientes. Por último, la programación genética, fue pensada para evolucionar programas de ordenador, representándolos en forma de árboles [6].

Por otro lado, los dominios en los que se aplica la computación evolutiva [7] abarca en la medicina, con la identificación de patologías en pacientes con riesgo de sufrir alguna de ellas; en biología y/o bioingeniería, análisis de secuencia de genes, de proteínas; en aplicaciones financieras, detección de fraude, de gastos en tarjetas de crédito; telecomunicaciones, descubrimiento de patrones de llamadas, detección de fraude; y en análisis de mercado y comercio, análisis de patrones de compra, evaluación de campañas

Digital Object Identifier (DOI): http://dx.doi.org/10.18687/LACCEI2017.1.1.329
ISBN: 978-0-9993443-0-9
ISSN: 2414-6390

publicitarias y segmentación de clientes. También se incluyeron otras aplicaciones [5] en sistemas CAD, identikit, programación de transporte de múltiples productos por oleoducto.

En investigaciones recientes [8] se encontró que las predicciones acerca del grado de incumplimiento de pago de clientes de bancos, es la esencia de la gestión de riesgo de crédito; para tener conocimiento de ello, la mayoría de bancos, calcula ese grado, en base a la agrupación de deudores con características específicas; pero según el artículo [1], es mejor calcular la capacidad de pago de los prestatarios, de manera individual; aplicando técnicas de minera de datos como son la regresión logística, redes bayesianas, y máquinas de soporte de vectores.

Un caso similar, en otro estudio, realizado acerca de las instituciones de microfinanzas en Perú; las aplicaciones de sistemas de puntuación de créditos bancarios son recientes, y además de la poca información que existe acerca de modelos estadísticos no paramétricos. En este artículo [9], se estudian modelos con el enfoque de perceptrón multicapa, como los de análisis discriminante lineal, discriminante cuadrático, y la regresión logística, concluyendo que la aplicación de redes neuronales tiene mucho potencial en cuanto a resultados sobre créditos bancarios se refiere.

En ese contexto, el presente trabajo, utiliza el enfoque de aprendizaje perezoso con el algoritmo de *k*-vecino más cercano (KNN), que según [10], se basa en la búsqueda de un conjunto de prototipos de *k* más cercanos al modelo a clasificar, por ello las predicciones se realizan en base a ejemplos similares al que se tiene que predecir. Lo datos de experimentación corresponden al incumplimiento de pago de la deuda de tarjetas de crédito de clientes de un banco Taiwán, Japón, en el año 2005.

El objetivo de este artículo es la predicción de incumplimiento de pago de clientes de tarjetas de crédito con el algoritmo del *k*-vecino más cercano, optimizado con un método de evaluación estadístico.

Este artículo está organizado de la siguiente manera: en la sección II describe el estado del arte, la sección III presenta los materiales y métodos, donde se explica el método a utilizar, además se detalla el modelo de operación evolutivo. En la sección IV se reportan los resultados obtenidos luego de la experimentación, finalmente en la sección V se presenta la conclusión con el grado de cumplimiento del objetivo de este trabajo.

II. ESTADO DEL ARTE

Dentro del estado del arte consultado se pasa a describir las investigaciones relacionados al tema de estudio.

Yeh et al. [11] en su investigación se enfoca en el caso de incumplimiento de los pagos de los clientes en Taiwán y compara la precisión predictiva de la probabilidad de incumplimiento entre seis métodos de minería de datos, entre los que destaca el clasificador del *k*-vecino más cercano (KNN), Regresión logística, análisis discriminante, clasificador bayesiano, redes neuronales y árboles de decisión. Desde la perspectiva de la gestión de riesgos, el resultado de la precisión predictiva de la probabilidad estimada de incumplimiento será más valiosa que el resultado binario de la clasificación - clientes creíbles o no creíbles. Debido a que la probabilidad real de incumplimiento es desconocida, este estudio presentó un "Método Suavizado de Clasificación" para estimar la probabilidad real de incumplimiento. Con la probabilidad real de incumplimiento como variable de respuesta (Y), y la probabilidad predictiva de incumplimiento como variable independiente (X), el resultado de regresión lineal simple (Y = A + BX) muestra que el modelo de predicción producido por la red neuronal artificial tiene el mayor coeficiente de determinación; Su intercepción de regresión (A) es cercana a cero, y el coeficiente de regresión (B) a uno. Por lo tanto, entre las seis técnicas de minería de datos, la red neuronal artificial es la única que puede estimar con precisión la probabilidad real de incumplimiento.

Respecto a técnicas de algoritmos evolutivos, Lavado et al. [12], revela datos que representan peligro y alerta a consumidores y negocios en Perú sobre el uso de tarjetas de débito, debido a incidentes de fraude, para lo cual plantea, el uso de un modelo de evaluación del comportamiento transaccional de clientes y sus patrones transaccionales, las cuales podrían ser tipificadas como sospechosas, por ello se aplicará técnicas de algoritmos genéticos; esto último para la maximización de predicción en contraste con datos reales, además de la minimización de falsos positivos y negativos que evalúa un sistema de detección de fraude. Dicho modelo, sigue un enfoque denominado *Iterative Rule Learning*, definiendo a cada individuo como una regla, y a otro como una posible solución. Cada regla es representada como un cromosoma, este último expresado en cadenas binarias de bit, y de acuerdo a una combinación de estos, se puede determinar si un individuo es sospechoso o no de haber cometido fraude. Los resultados de las operaciones del algoritmo se mostraron tanto para el canal POS (punto de venta), como para el canal internet. Para el primero, de un conjunto de 240 transacciones (200 transacciones genuinas y 20 fraudulentas), se logró un 95.8% de precisión, así, 211 de las 200 operaciones son VP (Verdaderos positivos), y 19 de las 20 operaciones fraudulentas son VN (Verdaderos Negativos). Para el canal internet, se alcanzó una precisión del 95,5%, dando como valores VP a 142 de 150 transacciones genuinas, y VN, a 51 de 52 fraudulentas.

En el trabajo de Aya et al [13], se plantea un modelo híbrido de clasificación basado en algoritmos genéticos (AG) y máquinas de vectores de soporte (SVM) aplicado a la evaluación crediticia. Para la especificación de la SVM, se utilizó el truco de Kernel, el cual consiste en encontrar un espacio de características de dimensión mayor al espacio de atributos en donde se puedan separar linealmente un conjunto de datos de otro. Para la especificación del AG, se hizo la codificación de cromosomas y la información generada por

aplicación de SVM se almacena. Para el mecanismo de selección se acudió al método *Stochastic Universal Sampling*, el cual provee de un sesgo en cero, y un mínimo esparcimiento en cuanto a los valores de falsos positivos.

Los resultados de la propuesta fueron que la aplicación del modelo híbrido GA-SVM mostró probabilidades de cruce y mutación en 50% y 60% respectivamente, mientras que para las de mutación fueron de 1%, 3% y 10%. Con lo anterior, se redujo la cantidad de falsos positivos, de esa manera se disminuye el riesgo en la aprobación de créditos.

Por otro lado, en la aplicación de técnicas de computación evolutiva, se tiene el trabajo de Gómez et al. [14], que estudia el pronóstico de calificación y riesgos crediticios en el sector público de México para el año 2009, realizando una comparación de tres técnicas como son: redes neuronales artificiales o RNA (para un mejor pronóstico de datos dentro y fuera de la muestra), modelo *Probit* ordenado, y análisis discriminante (genera una mejor estimación por intervalos de datos). El conjunto de datos usado fue provisto por una calificadora de la sección de finanzas públicas de ese país, dicho conjunto tuvo una muestra de 112 registros de concesión de créditos, para la medición de riesgo A+ descrito como alta capacidad de pago. Entre los resultados que se extraen están: la aplicación de RNA, arrojó aciertos del 100% de datos dentro de la muestra, teniendo a esta técnica como la más consistente. El modelo Probit, dió un 88,65 de aciertos, y para el análisis discriminante un 65,7%.

En otra investigación, acerca de la detección de fraude en tarjetas de crédito, Bentley et al [15] describió el uso de un sistema evolutivo difuso con la capacidad de clasificar las transacciones de tarjetas de crédito en sospechosas, y no sospechosas, valiéndose también de la programación genética; además, dicho sistema es capaz de alcanzar buenos niveles de precisión e inteligibilidad de datos reales. El modelo darwiniano difuso utilizado, soporta dos tipos de interpretación de reglas: lógica difusa tradicional, y lógica difusa de pertenencia-preservación.

Otra aplicación de algoritmos evolutivos, es lo presentado por Chávez et al [16], que utiliza la optimización por enjambre de partículas (PSO) junto a redes bayesianas (RB) para el diagnóstico de la hipertensión arterial (HTA) de habitantes de la ciudad de Santa Clara, Cuba, en el año 2009. PSO se basa en el comportamiento colectivo de individuos como aves, peces, entonces, esa característica se aplica para la búsqueda de la estructura de dicha red. El conjunto de datos describe a una muestra de 849 personas entre 18 y 78 años de edad. Los resultados del algoritmo son: para búsqueda global se encontró 2 padres, 40 partículas y, 1000 iteraciones; con ello se obtiene buenos modelos de RB de clasificación de HTA, con una exactitud del 95% de buena clasificación.

III. Materiales y Métodos

El presente estudio utiliza la base de datos que se recuperó del repositorio libre de UCI Machine Learning [17]. Los datos corresponden al incumplimiento de pago de la deuda de tarjetas de crédito de clientes de un banco importante en Taiwán, registros que datan del año 2005 [11]; se utilizó la base de datos referido porque refleja un modelo de comportamiento típico en el caso de estudio.

El archivo de origen de datos consta de 25 atributos, un total de 30000 instancias; del cual se extrajo una muestra con 24 atributos, y 5439 registros. Respecto a los atributos, estos tratan acerca del estado de los pagos realizados, en relación con el monto depositado; teniendo un atributo de predicción de incumplimiento de pago del siguiente mes.

Para la fase de entendimiento, se anexaron los enunciados de cada una de las 25 variables:

- *Incumple_pago_sig_mes*: probabilidad de incumplimiento de pago (1 = Si; 0 = No).
- *Monto_cred_dado*: Monto de crédito otorgado (expresado en Nuevos Dólares de Taiwan); lo cual incluye el crédito de consumo individual, y el crédito familiar.
- *Sexo*: (Varón = 1; Mujer = 2).
- Educación: (1 = Graduado; 2 = universitario; 3 = colegio; 4 = otros).
- *Estado_civil*: (1 = casado; 2 = soltero; 3 = otros).
- *Edad*: en años.
- *EstPagoSet05* - EstPagoAbr05: Historial de pagos anteriores; se registran pagos de meses anteriores (de abril a setiembre) como sigue: EstPagoSet05 = estado de pago en setiembre, 2005; EstPagoAgo05 = estado de pago en agosto, 2005;…; EstPagoAbr05 = estado de pago en abril, 2005. El intervalo de medición para ello es: -1 = pago a tiempo; 1 = pago retrasado por un mes; 2 = pago retrasado por dos meses;…; 8 = pago retrasado por ocho meses; 9 = pago retrasado por nueve meses.
- *EstCuentaSet05* - EstCuentaAbr05: monto del estado de cuenta. EstCuentaSet05 = monto de estado de cuenta en setiembre, 2005; EstCuentaAgo05 = monto de estado de cuenta en agosto, 2005;…; EstCuentaAbr05 = monto de estado de cuenta en abril, 2005.
- *MontoPagoSet05* - MontoPagoAbr05: monto de pago previo. MontoPagoSet05 = monto de pago previo en setiembre, 2005; MontoPagoAgo05 = monto de pago previo en agosto, 2005; …; MontoPagoAbr05 = monto de pago previo en abril, 2005.

Para la realización de la experimentación se utilizó el producto software *Keel* (*Knowledge Extraction based on Evolutionary Learning*) [18], que es una herramienta para uso y construcción de modelos en los campos de algoritmos de aprendizaje evolutivo y minería de datos como regresión, clasificación, aprendizaje no supervisado, etc.

Keel, tiene las siguientes ventajas [19], reducción del trabajo de programación, ya que incluye una librería de algoritmos de aprendizaje evolutivo, y simplificando la integración de tales algoritmos con diferentes técnicas de preprocesamiento. No se requiere de alto nivel de

conocimientos y experiencia de los investigadores, ya que está presente la librería antes mencionada, para poder aplicarla en problemas de investigación. Al ejecutarse sobre la máquina virtual Java, *Keel* se hace independiente de cualquier plataforma.

Para el procesamiento del conjunto de datos, se aplica el algoritmo de vecinos más cercanos (KNN), el cual clasifica nuevas instancias como la clase mayoritaria de entre los *k* vecinos más cercanos de entre los datos de entrenamiento, se rige bajo un modelo local, mas no existe un modelo global. Durante el proceso de entrenamiento, solo se guardan las instancias, no se construye modelo alguno, entonces, la clasificación se realiza al llegar la fase de evaluación.

Algunas limitaciones de este algoritmo son: alta sensibilidad a los atributos irrelevantes, al ruido; se torna lento si existe gran cantidad de datos de entrenamiento, además también depende de la función de distancia adecuada como se define en Ec. (1), Ec. (2) y Ec. (3).

Para la selección de *k*, se puede hacer validación cruzada de los datos de operación, se selecciona una familia o un tipo de modelo. Para la selección de instancias, existen dos técnicas:

- *Editing*: eliminar instancias engañosas si son clasificadas incorrectamente, esta acción la realizan los *k* vecinos de esas instancias, excepto en el interior de una clase; funciona si no hay presencia de demasiado ruido. Ésta técnica tiene por nombre *Wilson editing*.
- *Condensación*: eliminar instancias superfluas, pero mantiene los datos con ruido; si determinada instancia aún no está bien clasificada respecto a las existentes, se almacena. Ésta técnica lleva por nombre *Condensed Nearest Neighbor* (CNN). El funcionamiento de CNN depende en mayor grado del orden en que se toman las instancias. Una variante de esta forma de operación, es *Reduced Nearest Neighbor rule*, esta forma permite eliminar instancias ruido.

A continuación, se muestra el pseudocódigo de KNN, ver Fig. 1.

COMIENZO
  Entrada: $D = \{(\mathbf{x}_1, c_1), \ldots, (\mathbf{x}_N, c_N)\}$
    $\mathbf{x} = (x_1, \ldots, x_n)$ nuevo caso a clasificar
  PARA todo objeto ya clasificado (x_i, c_i)
    calcular $d_i = d(\mathbf{x}_i, \mathbf{x})$
  Ordenar $d_i (i = 1, \ldots, N)$ en orden ascendente
  Quedarnos con los K casos $D_{\mathbf{x}}^K$ ya clasificados más cercanos a $\mathbf{x}$
  Asignar a $\mathbf{x}$ la clase más frecuente en $D_{\mathbf{x}}^K$
FIN

Fig. 1. Pseudocódigo de KNN.

Para la selección del valor *k*, se puede utilizar alguna de las siguientes ecuaciones.

- Ecuación de la distancia Euclídea:

$$d(A,B) \equiv \sqrt{\sum_{i=1}^{n}(A_i - B_i)^2} = \sqrt{(A-B)^T(A-B)} \quad (1)$$

- Ecuación de la distancia Euclídea ponderada:

$$d(A,B) \equiv \sqrt{(A-B)^T M^T M (A-B)} \quad (2)$$

- Ecuación de la distancia de Manhattan:

$$d(x,y) = \sum_{i=1}^{n} |x_i - y_i| \quad (3)$$

Para la gestión de datos en *Keel*, se sigue los siguientes pasos:

- Importación de *dataset*: la herramienta acepta múltiples formatos de archivo, entre los principales están: .csv, .xls, .arff. El conjunto de datos para este trabajo, tuvo extensión .csv.
- Selección del tipo de problema: para este caso, se aplicó por clasificación, ya que KNN pertenece a esa categoría.
- Generación de particiones, que por defecto *Keel*, divide el d*ataset* en 10 particiones.
- Visualización de gráficos, a partir de la apertura de alguna partición, como se observa en la Fig. 2, la edad es determinante en el cumplimiento de pago de un préstamo bancario, además se nota que, a mayor cantidad de dinero prestado, se da un menor grado de impago.

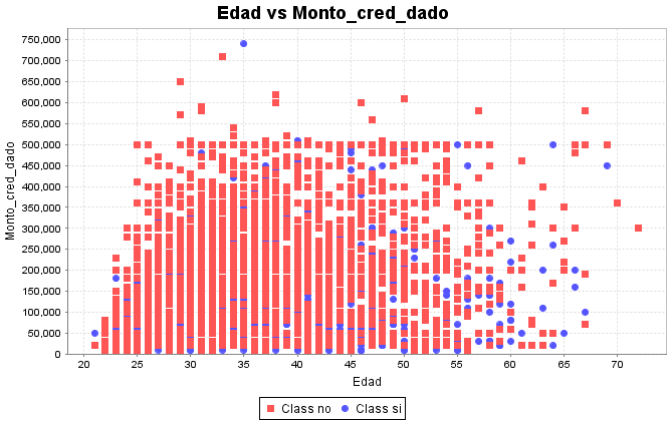

Fig. 2. Gráfico 2D, atributos edad y monto de crédito asignado.

Para la fase de experimentación de datos, junto con la herramienta *Keel*, se plantea el modelo de experimento mostrado en la Fig. 3.

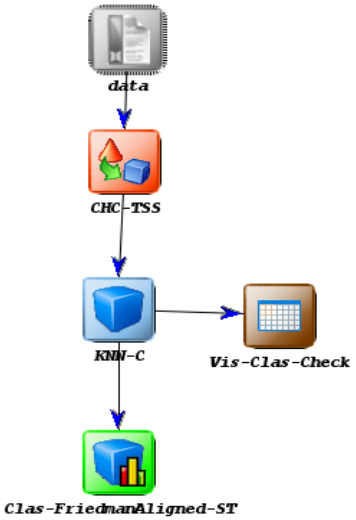

Fig. 3. Modelo de experimento en Keel.

El modelo de experimento, se compone de los siguientes elementos:

- *Conjunto de datos de entrada*: atributos relacionado al incumplimiento de pagos.
- *Preproceso*: algoritmo de selección del conjunto de entrenamiento evolutivo, de nombre CHC-TSS (CHC *Adaptative Search for Instance Selection*). Al tener un *dataset* con muchos registros, a veces no es posible trabajar con una cantidad inmensa de ellos, para ello, se utilizan mecanismo que permiten reducir el número de registro o columnas de un *dataset*. CHC-TSS [20], es un modelo que introduce diferentes características para obtener un equilibrio entre explotación y exploración de datos. La manera de trabajo de este esquema junto con los algoritmos genéticos es: utilizar una población de padres que generen individuos, para que estos últimos se emparejen y generen hijos potenciales; luego, como en el comportamiento de los seres vivos, se realice la competición por la supervivencia, en la que se seleccionen los mejores cromosomas de la población de padres y descendientes, para formar la siguiente generación.
- *Método o algoritmo*: el clasificador de aprendizaje perezoso KNN-C (*K-Nearest Neighbors Classifier*).
- *Método de evaluación estadístico*: algoritmo Clas-FriedmanAligned-ST (*Friedman Aligned Test and Post-Hoc Procedures*). Este modelo define una evaluación no paramétrica para comparaciones múltiples; lo que consiste en saber si en un conjunto de *k* muestras, al menos dos de las dos muestras son significativamente diferentes.
- *Muestra de resultados*: método *Vis-Clas-Check* para mostrar el resumen de los resultados del modelo de experimento, éstos se muestran en la sección IV.

En la fase de ejecución de experimentación, se trabaja con los archivos generados por la herramienta *Keel*. Para este paso, se utilizó el siguiente equipo:

- Hardware: Sistema operativo Windows 7, velocidad del procesador 3.84Ghz, capacidad de memoria: 8 GB.
- Software: componente Java Runtime Environment 8.0, Keel versión 3.

Como se aprecia en la ejecución del modelo de experimento en la Fig. 4, se muestra el procesamiento de los atributos de entrada junto con el atributo de salida de incumplimiento de pago, con el número de instancias leídas correctamente y la carga del archivo de forma correcta.

IV. Resultados

Los resultados obtenidos, luego de la ejecución del experimento con el modelo, como se muestran en las Fig. 3 y Fig. 4, se dan a conocer en la Fig. 5.

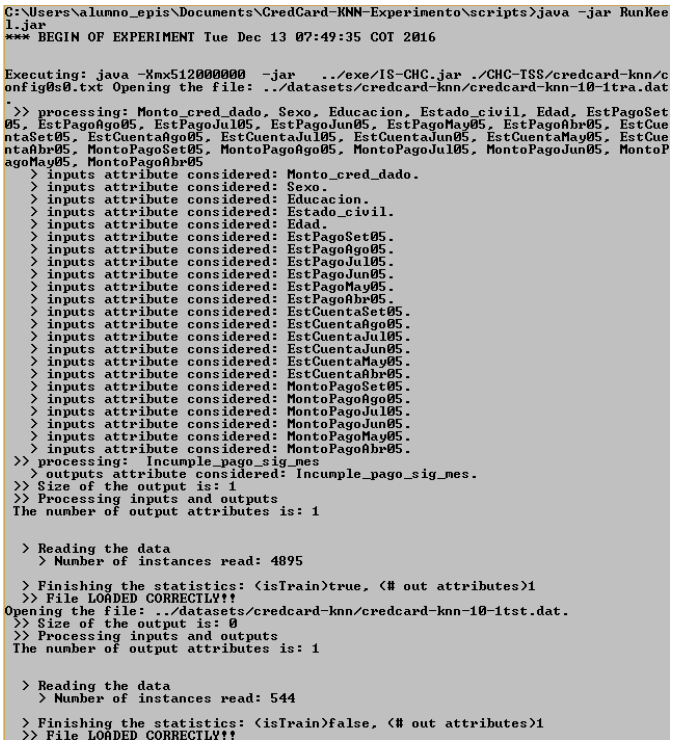

Fig. 4. Ejecución de script de experimento.

```
TEST RESULTS
============
Classifier= credcard-knn
Fold 0 : CORRECT=0.8492647058823529 N/C=0.0
Fold 1 : CORRECT=0.8308823529411764 N/C=0.0
Fold 2 : CORRECT=0.8198529411764706 N/C=0.0
Fold 3 : CORRECT=0.8713235294117647 N/C=0.0
Fold 4 : CORRECT=0.8308823529411764 N/C=0.0
Fold 5 : CORRECT=0.8088235294117647 N/C=0.0
Fold 6 : CORRECT=0.8161764705882353 N/C=0.0
Fold 7 : CORRECT=0.8198529411764706 N/C=0.0
Fold 8 : CORRECT=0.8327205882352942 N/C=0.0
Fold 9 : CORRECT=0.8287292817679558 N/C=0.0
Global Classification Error + N/C:
0.16914913064673381
stddev Global Classification Error + N/C:
0.01711009960614139
Correctly classified:
0.8308508693532661
Global N/C:
0.0
```

Fig. 5. Resultados de evaluador Clas- FriedmanAligned-ST junto con KNN-C.

Se hizo la comparación de resultados obtenidos, como se muestra en la Fig. 5, con lo que se obtiene al aplicar el clasificador KNN en la herramienta *Weka* (ver Fig. 6). El objetivo de este trabajo, fue optimizar la predicción de incumplimiento de pago de clientes de tarjetas de crédito. Con el uso de métodos evolutivos y el algoritmo KNN, en *Keel*, se obtuvo una clasificación correcta de instancias al 83%, sin embargo, al utilizar técnicas de minería de datos y el método KNN en *Weka*, se obtiene una clasificación correcta del 75%. Por lo tanto, se entiende que la computación evolutiva ayuda de mejor manera a las instituciones financieras a medir el

riesgo de préstamos de crédito, con el fin de clasificar a clientes que verdaderamente merecen la concesión, y denegar a personas supuestamente morosas.

```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances        4083               75.0689 %
Incorrectly Classified Instances      1356               24.9311 %
Kappa statistic                          0.3667
Mean absolute error                      0.2493
Root mean squared error                  0.4991
Relative absolute error                 64.1469 %
Root relative squared error            113.2167 %
Total Number of Instances             5439
```

Fig. 6. Resultados para KNN, en Weka.

La matriz de confusión obtenida por la herramienta *Keel* se muestra en la Fig. 7, y para *Weka*, en la Fig. 8. Por consiguiente, se calcula: sensitividad o precisión positiva (*S*, Ec. (4)), especificidad o precisión negativa (*E*, Ec. (5)) y precisión (*P*, Ec. (6)). Los valores de dicha matriz son: VP (verdaderos positivos), VN (verdaderos negativos), FP (falsos positivos), FN (falsos negativos).

$$S = \frac{VP}{VP+FN} \tag{4}$$

$$E = \frac{VN}{VN+FP} \tag{5}$$

$$P = \frac{VP+VN}{VP+VN+FP+FN} \tag{6}$$

```
Set:test
Total percentage of successes:
0.83
Percentage of successes in each partition:
1       0.849
2       0.83
3       0.819
4       0.871
5       0.83
6       0.808
7       0.816
8       0.819
9       0.832
10      0.828
Confusion matrix (rows=real class;columns=obtained class):
3598    405
515     921
```

Fig. 7. Matríz de confusión generada por *Keel*.

Cálculo de valores para *Keel*: *S*= 0.8988, *E*=0.6414, *P*=0.8309

```
    a    b   <-- classified as
 3296  707 |    a = no
  649  787 |    b = si
```

Fig. 8. Matríz de confusión generada por *Weka*.

Cálculo de valores para *Weka*: *S*=0.8234, *E*=0.5481, *P*=0.7507

V CONCLUSIONES

En este estudio se analizó la problemática del incumplimiento de pago de clientes con tarjetas de crédito en las entidades financieras mediante la aplicación de un modelo experimental del algoritmo k-vecino más cercano y Clas-FriedmanAligned-ST.

Los resultados experimentales de la investigación son alentadores mostrando que el modelo empleado es capaz de alcanzar una buena precisión en la predicción de clientes y las deudas de cada uno según los valores obtenidos de sensitividad *S*= 0.8988, especificidad *E*=0.6414, y precisión *P*=0.8309 que representa el 83%, a comparación de la aplicación de técnicas de minería de datos con poca precisión del 75%, por lo tanto, esta es una estrategia apropiada que minimiza el riesgo esperado respecto a la predicción de incumplimiento de pago de clientes con tarjetas de crédito.

REFERENCIAS

[1] A. I. Tudor, A. Bâra, and S. V. Oprea, "Comparative analysis of data mining methods for predicting credit default probabilities in a retail bank portfolio," in *WSEAS International Conference. Proceedings. Recent Advances in Computer Engineering Series*, 2012, no. 7.

[2] D. J. Hand and W. E. Henley, "Statistical Classification Methods in Consumer Credit Scoring: a Review," *J. R. Stat. Soc. Ser. A (Statistics Soc.*, vol. 160, no. 3, pp. 523–541, 1997.

[3] H. C. Koh and C. K. L. Gerry, "Data mining and customer relationship marketing in the banking industry," *Singapore Manag. Rev.*, vol. 24, no. 2, pp. 1–27, 2002.

[4] E. Arrese Jiménez, "Estimación de intervalos de predicción mediante computación evolutiva," 2015.

[5] H. Banda, "Inteligencia Artificial: Principios y Aplicaciones." Obtenido de Academia. edu: https://goo. gl/jg4E57, 2014.

[6] N. L. Cramer, "A representation for the adaptive generation of simple sequential programs," in *Proceedings of the First International Conference on Genetic Algorithms*, 1985, pp. 183–187.

[7] A. Villagra, D. Pandolfi, M. G. Lasso, M. E. de San Pedro, and G. Leguizamón, "Algoritmos evolutivos y su aplicabilidad en la tarea de clasificación," in *VIII Workshop de Investigadores en Ciencias de la Computación*, 2006.

[8] S. K. Jena, A. Kumar, and M. Dwivedy, "Banking Credit Scoring Assessment Using Predictive K-Nearest Neighbour (PKNN) Classifier," *Handb. Res. Intell. Tech. Model. Appl. Mark. Anal.*, p. 332, 2016.

[9] A. Blanco, R. Pino-Mejías, J. Lara, and S. Rayo, "Credit scoring models for the microfinance industry using neural networks: Evidence from Peru," *Expert Syst. Appl.*, vol. 40, no. 1, pp. 356–364, 2013.

[10] C. García and I. Gómez, "Algoritmos de aprendizaje: KNN & Kmeans." Documento, 2012.

[11] I.-C. Yeh and C. Lien, "The comparisons of data mining techniques for the predictive accuracy of probability of default of credit card clients," *Expert Syst. Appl.*, vol. 36, no. 2, pp. 2473–2480, Mar. 2009.

[12] L. E. L. Napaico, "Un Algoritmo genético para la detección de fraude electrónico en tarjetas de débito en el Perú," *Rev. Investig. Sist. e Informática*, vol. 10, no. 1, pp. 87–97, 2013.

[13] R. Aya, A. Yesid, and others, "Modelo híbrido de clasificación basado en algoritmos genéticos y máquinas de vectores de soporte aplicado a la evaluación crediticia," Universidad Nacional de Colombia, 2010.

[14] P. Gómez and A. Mendoza, "Herramientas para el pronóstico de la calificación crediticia de las finanzas públicas estatales en México: redes neuronales artificiales, Modelo PROBIT ordenado y análisis discriminante," *Com. Investig. del Premio Nac. Mercados Financ. la*

Bols. Mex. Valores, pp. 1–37, 2009.
[15] P. J. Bentley, J. Kim, G.-H. Jung, and J.-U. Choi, "Fuzzy darwinian detection of credit card fraud," in *the 14th Annual Fall Symposium of the Korean Information Processing Society, 14th October*, 2000.
[16] M. del Carmen Chávez, G. Casas, J. Moreira, E. González, R. Bello, and R. Grau, "Uso de redes bayesianas obtenidas mediante Optimización de Enjambre de Partículas para el diagnóstico de la Hipertensión Arterial," *Investig. Operacional*, vol. 30, no. 1, pp. 52–61, 2009.
[17] K. Bache and M. Lichman, "UCI Machine Learning Repository," *University of California Irvine School of Information*, vol. 2008, no. 14/8. p. 0, 2013.
[18] J. Alcalá-Fdez, M. J. del Jesus, J. M. Garrell, F. Herrera, C. Herbás, and L. Sánchez, "Proyecto KEEL: Desarrollo de una herramienta para el análisis e implementación de algoritmos de extracción de conocimiento evolutivos," *Tendencias la Minería Datos en España, Red Española Minería Datos y Aprendiz.*, pp. 413–424, 2004.
[19] J. Alcalá-Fdez *et al.*, "KEEL: a software tool to assess evolutionary algorithms for data mining problems," *Soft Comput.*, vol. 13, no. 3, pp. 307–318, 2009.
[20] J. R. Cano, F. Herrera, and M. Lozano, "Using evolutionary algorithms as instance selection for data reduction in KDD: An experimental study," *IEEE Trans. Evol. Comput.*, vol. 7, no. 6, pp. 561–575, 2003.