



UNIVERSIDAD TECNICA PARTICULAR DE LOJA

La Universidad Católica de Loja

ÁREA DE INFORMÁTICA

TITULO DE INGENIERO EN INFORMÁTICA

Desarrollo de un modelo predictivo para detección de riesgo crediticio

utilizando redes neuronales para una entidad financiera.

TRABAJO DE TITULACIÓN

AUTOR: Gaona Martínez, Romel Alejandro.

DIRECTOR: Riofrío Calderón, Guido Eduardo, Ing.

CENTRO UNIVERSITARIO QUITO

2016

APROBACIÓN DEL DIRECTOR DEL TRABAJO DE TITULACIÓN

Ingeniero.

Guido Eduardo Riofrío Calderón.

DOCENTE DE LA TITULACIÓN

De mi consideración:

El presente trabajo de titulación: Desarrollo de un modelo predictivo para detección de riesgo crediticio utilizando redes neuronales para una entidad financiera, realizado por Gaona Martínez Romel Alexandro, ha sido orientado y revisado durante su ejecución, por cuanto se aprueba la presentación del mismo.

Loja, enero de 2016

f)

DECLARACIÓN DE AUTORÍA Y CESIÓN DE DERECHOS

“Yo, Gaona Martínez Romel Alexandro declaro ser autor del presente trabajo de titulación: Desarrollo de un modelo predictivo para detección de riesgo crediticio utilizando redes neuronales para una entidad financiera, de la Titulación de Ingeniero en Informática, siendo el Ing. Guido Eduardo Riofrío Calderón director del presente trabajo; y eximo expresamente a la Universidad Técnica Particular de Loja y a sus representantes legales de posibles reclamos o acciones legales. Además certifico que las ideas, conceptos, procedimientos y resultados vertidos en el presente trabajo investigativo, son de mi exclusiva responsabilidad.

Adicionalmente declaro conocer y aceptar la disposición del Art. 88 del Estatuto Orgánico de la Universidad Técnica Particular de Loja que en su parte pertinente textualmente dice: “Forman parte del patrimonio de la Universidad la propiedad intelectual de investigaciones, trabajos científicos o técnicos y tesis de grado o trabajos de titulación que se realicen con el apoyo financiero, académico o institucional (operativo) de la Universidad”.

f).

Autor: Gaona Martínez Romel Alexandro

Cédula: 1103150056

DEDICATORIA

El presente trabajo va dedicado a mi familia y amigos que me ayudaron en la culminación del mismo, ya que con su ayuda, consejos y sugerencias hicieron posible que se lleve a cabo satisfactoriamente.

También va dedicado a aquellas personas que siempre están en busca del conocimiento o siempre ven la necesidad de mejorar algo, con lo cual siempre estarán aportando para el mejoramiento de nuestra sociedad y el mundo en general.

AGRADECIMIENTO

Agradezco sinceramente a la Universidad Técnica Particular de Loja, por ser una institución seria en donde pude educarme y adquirir conocimientos, los cuales me permiten desempeñarme en mi vida laboral.

Agradezco al personal de la institución financiera, ya con su apoyo que me permitieron realizar el presente trabajo.

Agradezco a las personas que con su experiencia aportaron con alguna idea en alguna parte de este trabajo de investigación.

INDICE DE CONTENIDOS

CARÁTULA.....	i
CERTIFICACIÓN.....	ii
DECLARACION DE AUTORÍA Y CESIÓN DE DERECHOS.....	iii
DEDICATORIA.....	iv
AGRADECIMIENTO.....	v
INDICE DE CONTENIDOS.....	vi
RESUMEN.....	1
ABSTRACT.....	2
INTRODUCCIÓN.....	3
PLANTEAMIENTO DEL PROBLEMA.....	5
JUSTIFICACION.....	6
ALCANCE.....	7
OBJETIVOS.....	8
CAPITULO 1. MARCO TEÓRICO.....	9
1.1. Introducción a la Inteligencia Artificial.	10
1.2. Clasificación y Predicción con IA.	11
1.3. Técnicas para clasificación y predicción.....	12
1.4. Data Mining.	12
1.5. Arboles de Decisión, Clasificación.	16
1.6. Redes neuronales artificiales.	18
1.7. El Perceptrón multicapa.	21
1.8. Ventajas y desventajas del uso de la IA.	24
CAPITULO 2. LA INTELIGENCIA ARTIFICIAL APLICADA AL RIESGO CREDITICIO.....	26
2.1. Concepto de riesgo crediticio.	27
2.2. Factibilidad de aplicación en procesos financieros.	28
2.3. Estudio del caso: entidad financiera en donde se aplicará el modelo de gestión de riesgos.	28
2.4. Variables a evaluar en el riesgo crediticio.	30
2.5. Resultados esperados aplicando IA.	32
CAPITULO 3. PLATAFORMA A UTILIZAR PARA EL TRATAMIENTO DE DATOS: WEKA.....	33
3.1. Introducción a Weka.	34
3.2. Ventajas de utilizar Weka.	34
3.3. Gestión de datos y tipos de variables.	35
3.4. Elaboración del archivo de datos para procesamiento.	36

CAPITULO 4. PROCESAMIENTO DE LA INFORMACIÓN EN WEKA.....	37
4.1. Ejecución e ingreso al programa.	38
4.2. Módulo Explorer.	39
4.3. Componentes principales del Explorer.	40
4.3.1. Funcionamiento de Preprocess.	40
4.3.2. Funcionamiento de Classify.....	44
4.4. Algoritmos de Clasificación en Weka.....	46
4.4.1. Algoritmo J48.	46
4.4.2. Perceptrón multicapa.	51
4.4.3. BayesNet y NaiveBayes.	52
4.4.4. Comparación de resultados de los algoritmos.	54
4.5. Selección de atributos relevantes.	55
4.6. Validación de resultados del modelo predictivo.	63
CAPITULO 5. IMPLEMENTACIÓN DEL MODELO PREDICTIVO.....	66
5.1. Presentación de la información.	67
5.2. Plataforma de desarrollo a utilizar: Java Netbeans, creación del proyecto.....	67
5.3. Conexión y manejo de procesos con Weka.	68
5.4. Interfaz de usuario para presentación de resultados.....	74
5.5. Aplicación del modelo predictivo desarrollado.	81
CONCLUSIONES.	84
RECOMENDACIONES.	85
BIBLIOGRAFIA.	86
ANEXOS.	87

RESUMEN

Con el presente trabajo se desea ayudar a minimizar los riesgos crediticios mediante la verificación del acceso de los clientes a los créditos que otorga la entidad financiera que es objeto de estudio; el objetivo es aprovechar la fortaleza de las herramientas de la computación e inteligencia artificial que nos pueden ayudar en ésta área, debido a que sus diversas técnicas han sido aplicadas en finanzas con exitosos resultados dentro del ámbito de detección y predicción.

El éxito de nuestra tarea se basa en presentar una guía y poder ayudar a la institución a verificar si el cliente es factible de crédito o no, de esa manera tanto el deudor como el acreedor no se verán inmersos en el futuro con posibles problemas de finanzas, tales como problemas legales y deudas. Con el resultado obtenido, la institución financiera podrá contar con una herramienta para evaluar datos y ayuda en la toma de decisiones, ya que siempre es bueno acompañar a la experiencia personal con la tecnología y nuevos procedimientos.

PALABRAS CLAVE: riesgos, clientes, créditos, computación, inteligencia artificial, técnicas, finanzas, predicción, herramienta, datos, tomas de decisiones, tecnología, procedimientos.

ABSTRACT

With this work we want to help minimize credit risk by verifying customer access to credit granted by the financial institution that is being considered; The goal is to leverage the strength of computer tools and artificial intelligence that can help us in this area, because their various techniques have been applied in finance with successful results within the field of detection and prediction.

The success of our work is based on presenting a guide and to help the institution to verify if the customer credit is feasible or not, that way both the debtor and the creditor will not be involved in the future with possible problems of finance such as legal problems and debts. With the result, the financial institution may have a tool to evaluate data and aid in decision-making, and it's always good to accompany personal experience with technology and new procedures.

KEYWORDS: risk, customers, credit, computer, artificial intelligence, technical, finance, forecasting, tool, data, decision making, technology, procedures.

INTRODUCCION

Nuestro trabajo de investigación consiste en desarrollar un modelo predictivo mediante un software o programa informático para la Predicción de Riesgos Crediticios al momento de seleccionar los clientes de una entidad financiera aprovechando las bondades de las herramientas de la Informática y de la Inteligencia Artificial

Para la implementación de nuestro trabajo vamos a utilizar información de los clientes que actualmente posee la empresa, dicha información se ingresa en el programa y se obtendrán resultados que serán evaluados; lo cual servirá de base para verificar si los futuros clientes son factibles de recibir el crédito.

Del primero hasta el tercer capítulo hacemos referencia a la parte teórica del trabajo, sobre conceptos de la Inteligencia Artificial y las técnicas y procedimientos dentro de ese campo que nos van a servir para nuestro fin. El cuarto capítulo se trata sobre conceptos de riesgo crediticio y sobre la factibilidad y bondades de la IA para tratar dichos temas; basándonos en el análisis de la información, el modelo a desarrollar y la herramienta de minería de datos en el quinto capítulo. Finalmente en el sexto capítulo presentaremos el modelo con los resultados esperados, utilizando el lenguaje Java como plataforma para el tratamiento y presentación de los mismos.

El problema principal a tratar es la elección adecuada de los clientes que son sujetos de crédito, para ello se tuvo que receptar la información de los clientes de la institución, las variables y los criterios de calificación que allí se manejan, lo cual nos sirvió de base para detectar los patrones que determinan si un cliente puede ser buen pagador o no.

Con los resultados que se lograron aplicando el modelo propuesto, se busca definir un proceso para la buena elección de los clientes, lo cual permita a la empresa manejar bien sus inversiones y en cuanto al cliente no incurrir en gastos que no pueda pagar y por ello incurrir en deudas no convenientes y problemas de morosidad; con ello se puede aportar a la sociedad con un grano de arena en el mejor manejo de los créditos y sus riesgos.

En cuanto al acceso de la información no se tuvo mayor inconveniente, ya que el personal de la cooperativa de ahorro y crédito prestó las facilidades necesarias del caso para el desarrollo e implementación del presente trabajo.

Como herramienta para el procesamiento de la información se utilizó el software Weka, en el cual se pudo realizar un análisis predictivo y minería de datos de todos los clientes; Para la clasificación de la información y selección de resultados se utilizaron alguna técnicas de la herramienta, tales como árboles de clasificación, redes neuronales, redes bayesianas, reglas de decisión, metaclasificadores, etc. Después del análisis de las técnicas se verificó el árbol de clasificación es el que mejor resultado presentó, con los resultados explicados en el trabajo; además se pudo comprobar que Weka es una herramienta ideal para trabajar con la clasificación de datos y predicción.

PLANTEAMIENTO DEL PROBLEMA.

Hoy en día, en el ámbito financiero, en especial en aquellas empresas en donde se maneja el crédito o préstamo de dinero a clientes, no se cuenta con la suficiente información o evidencia para determinar si una determinada persona es accesible a un crédito o no. Uno de los problemas con que se enfrentan las empresas financieras es tener la certeza de poder conceder un crédito a un cliente. Muchas de las veces las veces, el personal encargado de realizar ésta gestión dentro de la empresa, al no contar con las herramientas adecuadas, acude al empirismo o experiencia: con lo cual muchas de las veces se cae en malas decisiones.

Hay muchas entidades financieras que no utilizan buenas herramientas para minimizar el riesgo crediticio por falta de conocimiento, o por no querer hacer mejoras en su tecnología.

JUSTIFICACION.

El presente trabajo nació de la necesidad de bajar la tasa de riesgo al momento de elegir clientes para crédito, de una cooperativa de ahorro y crédito de la ciudad de Quito; se cuenta con herramientas convencionales y muchas del punto de vista personal, sin contar con una alternativa adicional que permita bajar la incertidumbre y ayudar en la toma de decisiones.

La probación de créditos es un tema bastante delicado dentro de las operaciones cotidianas de una entidad financiera, si no se tiene un análisis previo y correcta interpretación de resultados puede generarse con seguridad un riesgo financiero. Se pretende implementar la metodología de las redes neuronales artificiales en donde se puede aplicar la predicción, clasificación y presentación de resultados, aplicando una herramienta ampliamente probada para el efecto como Weka, la cual es un software accesible y fácil de adquirir; esta forma se podrá contar con datos alternativos que pueden ser comparados con los obtenidos de forma tradicional y comparar resultados.

ALCANCE.

La metodología de la inteligencia artificial y redes neuronales es factible a utilizar para la realización de éste trabajo pero también puede tener pequeños porcentajes de error en el entrenamiento de la red neuronal, ya que los resultados de salida no siempre son perfectos, pero si nos puede ayudar bastante en la toma de decisiones acertada según la tendencia de su resultado. El software Weka podrá ser implementado en un puesto de trabajo, y no se requiere con el mismo trabajar en red.

OBJETIVOS.

Objetivo General.

Implementar un modelo de predicción y detección de Riesgo Crediticio en la institución financiera, con ayuda de herramientas de la Inteligencia Artificial, con lo cual se podrá contar en la empresa con información adicional que será de gran ayuda para tratar actividades de crédito hacia los clientes.

Objetivos Específicos.

- Procesar y evaluar la información de los clientes dentro de la empresa y verificar su resultado.
- Realización de comparaciones de los resultados obtenidos con la red neuronal con los que se posee con medios convencionales.
- Ayudar a implementar una herramienta y metodología tecnológica que será de gran ayuda dentro de la empresa.
- Ayuda a verificar que información extra necesita la empresa para la evaluación de los riesgos.

CAPITULO I
MARCO TÉORICO

1.1. Introducción a la Inteligencia Artificial.

Se puede definir a la inteligencia artificial (IA) como un área que se encarga de crear y controlar entidades¹ que eligen las mejores opciones para resolver problemas por si mismas utilizando como referencia la inteligencia humana y una aproximación a la representación del conocimiento (Figura 1).

A continuación se presentan otros conceptos sobre la Inteligencia Artificial:

"La Inteligencia Artificial estudia cómo lograr que las máquinas realicen tareas que, por el momento, son realizadas mejor por los humanos" [Rich y Knight 1994]

"La parte de la Informática que trata de Sistemas computarizados inteligentes, es decir, Sistemas que muestran las características que pueden asociarse a la inteligencia en lo que se refiere al comportamiento humano: comprensión del lenguaje, aprendizaje, razonamiento, resolución de problemas, etc." [Amat 1989]

"El ofrecimiento por parte de la máquina de un comportamiento parecido al humano que es capaz de acomodarse o ajustarse a una disposición o situación real o ficticia y poder escoger de acuerdo a una serie de particularidades para dar una respuesta rápida y lo más acertada posible." [Costa 1995]

"La IA es el arte de crear maquinas con capacidad de realizar funciones que realizadas por personas requieren de inteligencia." [Kurzweil, 1990]

La Inteligencia Artificial tiene muchos campos de aplicación, entre los cuales podemos citar las siguientes:

- En Finanzas sirven para pronósticos financieros, riesgos de crédito, detección de fraudes.
- En Negocios, se puede usar en verificación de perfiles de mercados, de clientes, en campañas de ventas.
- Aprendizaje
- Sistemas Expertos, Robótica
- Descubrimiento de Información (data mining)
- Optimización de procesos industriales y control de calidad.
- Tratamiento de texto e imágenes.
- Aplicaciones médicas como diagnósticos, análisis de imágenes.
- Investigación científica.
- Predicción de consumos energéticos

¹ Una entidad puede ser un programa informático o hardware computacional

- Aplicaciones meteorológicas.
- Transporte y comunicaciones, optimización de tráfico y rutas.
- Bibliometría.
- Clasificación y Predicción, entre otros, siendo éste último nuestro ámbito a tratar dentro del presente trabajo de Investigación.

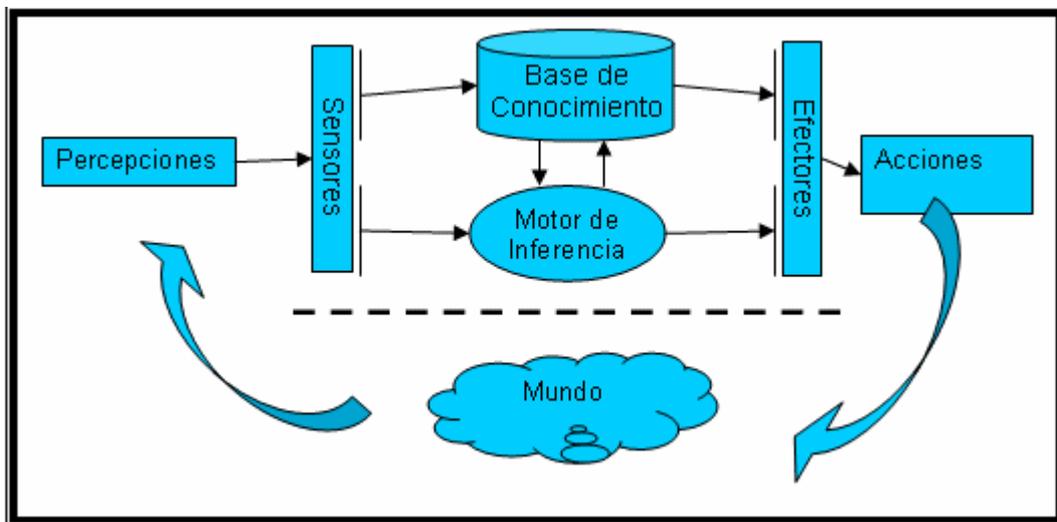


Figura 1. Representación del conocimiento

Fuente: <http://www.monografias.com/trabajos51/inteligencia-artificial/inteligencia-artificial.shtml>

1.2. Clasificación y Predicción con IA.

Según Mario Bunge (1979), las predicciones guardan relación con teorías y datos científicos, es decir, la predicción es una ampliación de la teoría científica. La predicción anticipa resultados en base al conocimiento, comparando la teoría con los resultados técnicos.

En cuanto a la Clasificación de Información y Predicción de resultados en el ámbito computacional, la Inteligencia Artificial ha tenido avances muy significativos y representa un aporte valioso para realizar éste tipo de tareas; por ejemplo, con técnicas que nos permiten encontrar información valiosa dentro una gran cantidad de datos sin tratar, con lo cual se nos hace más fácil la búsqueda de la información y en cuanto a la Predicción, podemos tener resultados precisos mediante el tratamiento de la información existente y anticiparnos con óptimos resultados. Una herramienta que se dispone hoy en día para la Clasificación y Predicción es la herramienta Weka², la cual vamos a utilizar para llevar a cabo el presente trabajo.

² Weka son las iniciales de Waikato Environment Knowledge Analysis

1.3. Técnicas para Clasificación y Predicción.

Existen algunas técnicas y procedimientos disponibles dentro de la Inteligencia Artificial para la Clasificación y Predicción tales como la Minería de Datos, Redes Neuronales, Reglas de Decisión, Árboles de Clasificación, Redes Bayesianas, algoritmos de aprendizaje supervisado entre otros. Dentro de nuestra herramienta a utilizar vamos a utilizar algunas técnicas hasta verificar cual nos va a brindar el mejor resultado en base a los datos reales que vamos a manejar.

Cada técnica es apropiada para cierto campo específico o tipo de información; entre las comúnmente conocidas están los Árboles de Clasificación y las Redes Neuronales, cada una de ellas tiene diferentes métodos o forma de operar, pero vamos a analizar cuales es más conveniente para nuestro fin. La Minería de datos es un procedimiento que sirve de gran ayuda a alguna técnica específica para encontrar información.

La mayoría de compañías tienen una gran cantidad de datos almacenados en sus ordenadores. Estos datos contienen una información que puede ser de gran utilidad para los resultados de la empresa. La gran abundancia de datos o su deficiente estructura puede hacer muy difícil extraer esta información útil. El objetivo de la clasificación de datos es la extracción de forma automática de información relevante, útil y no evidente contenida en dichos datos (data mining). Existen tres razones fundamentales por las cuales la búsqueda apropiada de la información es una realidad en nuestros días:

- Avances tecnológicos en almacenamiento masivo de datos y CPU.
- Existencia de nuevos algoritmos para extraer información en forma eficiente.
- Existencia de herramientas automáticas que no hacen necesario el ser un experto en estadística, redes neuronales, o algoritmos matemáticos para procesar de buena manera los datos, encontrar los resultados deseados y predecir resultados con eficacia.

1.4 . Data Mining.

Una definición tradicional es la siguiente: "*Un proceso no trivial de identificación válida, novedosa, potencialmente útil y entendible de patrones comprensibles que se encuentran ocultos en los datos*" (Fayyad y otros, 1996)

También se puede llamar al Data Mining o minería de datos (Figura 2) al conjunto de técnicas o procedimientos para extraer información relevante que se encuentra oculta dentro de una totalidad de datos para extraer la información, segmentarla, procesarla y poder

comprenderla; de ésta manera la Minería de Datos es un parte esencial en el descubrimiento de conocimientos dentro de un ámbito en donde no se puede con encontrar con facilidad los datos relevantes.

Inicialmente, de forma general todos los datos sin tratar son materia bruta sin utilidad; cuando se le encuentra un significado pasa a ser información y cuando ésta última nos presenta una representa una utilidad o una contribución se convierte en conocimiento.

Para realizar el Data Mining se hace necesario realizar las siguientes etapas, las cuales nos llevarán a encontrar dicho conocimiento esperado (Figura 3):

- **Selección de Objetivos:** se determina lo que se necesita al finalizar el estudio de la base de datos.
- **Selección y preprocesado de datos:** se realiza una depuración, transformación y reducción de la cantidad global de la cantidad bruta de los datos.
- **Selección de variables:** las cuales ayudarán a definir los criterios de clasificación de los datos.
- **Transformación de los datos:** se hace uso de métodos o algoritmos de la inteligencia artificial por medio de análisis y modelos para tener una visión de la información que se quiere obtener.
- **Extracción del conocimiento:** donde se va a seleccionar los datos más relevantes dentro la totalidad.
- **Interpretación y Evaluación de Resultados:** determinar si el resultado es el que se esperaba o se acerca a la realidad, con los cual representaría una utilidad o valor agregado para poder tomar decisiones.

Entre los diversos criterios para seleccionar los componentes de la Minería de Datos, según Joshi (1997) considera que "La minería de datos cuenta con tres grandes componentes según: Clustering³ o clasificación, reglas de asociación y análisis de secuencias" (p.82).

³ Clustering significa algoritmos de agrupamiento, pertenecientes al grupo de minería de datos no supervisados, cuyo fin es entender la relación entre objetos.

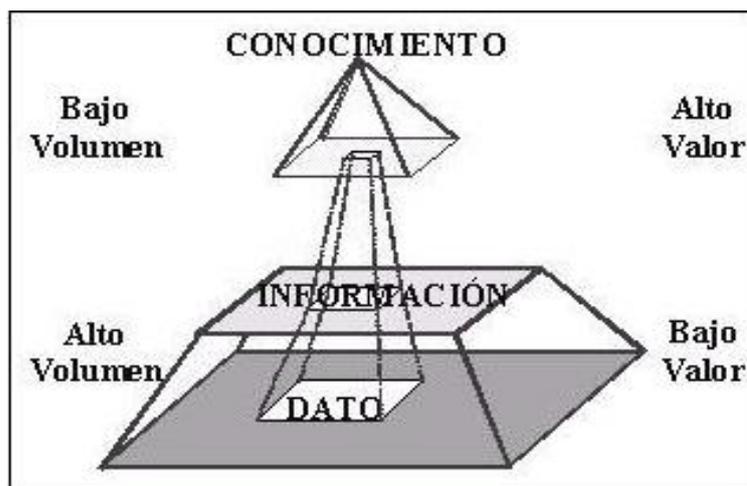


Figura 2. Minería de datos

Fuente: <https://sistemas2009unl.wordpress.com/data-mining-para-agronomia/>

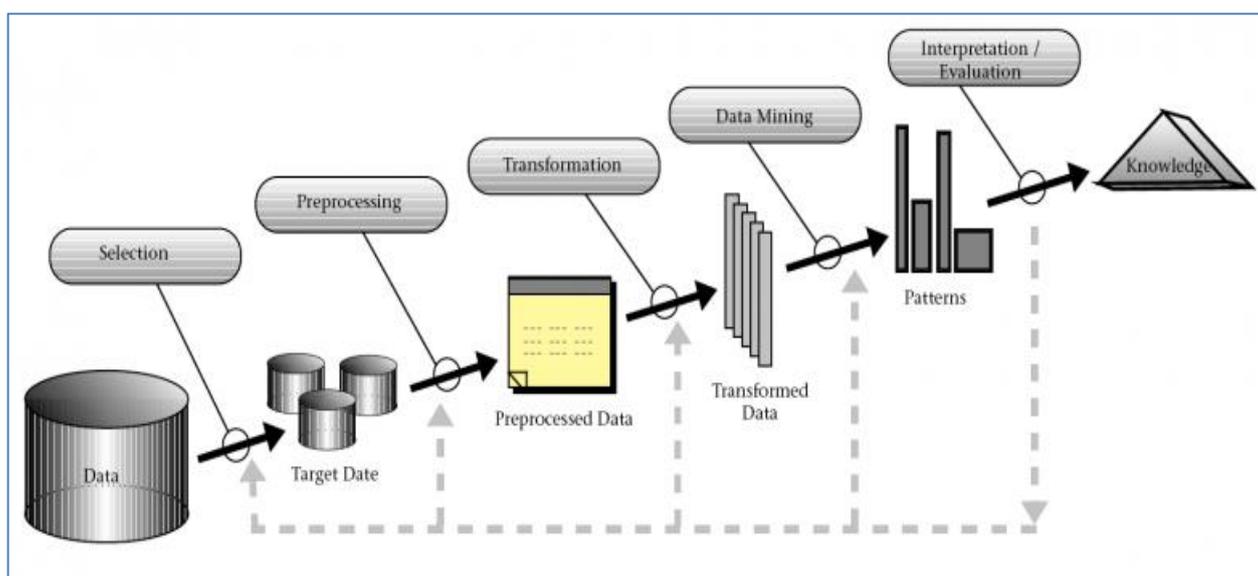


Figura 3. Etapas del Data Mining.

Fuente: <http://www.infovis-wiki.net/index.php?title=File:Fayyad96kdd-process.png>

La minería de datos busca generar información similar a la que podría producir un experto humano, que además satisfaga el Principio de Comprensibilidad, la minería de datos es el proceso de descubrir conocimientos interesantes, como patrones, asociaciones, cambios, anomalías y estructuras significativas a partir de grandes cantidades de datos almacenadas en bases de datos, data warehouses, o cualquier otro medio de almacenamiento de información.

Una vez comprendido el proceso de la minería de datos, un concepto interesante sobre el **descubrimiento de conocimientos** consiste en la iteración de los siguientes pasos:

- Limpieza de datos (Data cleaning), tratamiento de datos erróneos e irrelevantes.
- Integración de datos (Data integration), integración de datos en una única fuente.
- Selección de datos (Data selection), extracción de datos relevantes hacia al área de análisis.
- Transformación de datos (Data transformation), consolidación en formas apropiadas para la minería mediante procedimientos de agregación.
- Minería de datos, proceso esencial en donde por diversos métodos se extraen características comunes o patrones de los datos
- Evaluación de patrones (Pattern evaluation), identificación de patrones interesantes de acuerdo a algún parámetro impuesto por el usuario.
- Presentación de conocimientos (Knowledge evaluation), técnicas para representar los conocimientos adquiridos.

Siguiendo con la minería de datos, según Joshi, (1997). La clasificación de datos desarrolla una descripción o modelo para cada una de las clases presentes en la base de datos. Existen muchos métodos de clasificación como aquellos basados en los árboles de decisión TDIDT como el ID3 y el C4.5, los métodos estadísticos, las redes neuronales, y los conjuntos difusos, entre otros.

A continuación se describen brevemente aquellos métodos de aprendizaje automático que han sido aplicados a la minería de datos con cierto éxito:

- Algoritmos estadísticos.
- Redes Neuronales.
- Algoritmos genéticos.
- Método del vecino más cercano: es una técnica que clasifica cada registro de un conjunto de datos en base a la combinación de las clases de los k registros más similares. Generalmente se utiliza en bases de datos históricas.
- Clustering: son métodos de agrupación de datos que nos permiten clasificar los datos por su similitud entre ellos. Son utilizadas con frecuencia para entender los grupos naturales de clientes en empresas o bancos.
- Árboles de decisión: Son estructuras en forma de árbol que representan conjuntos de decisiones capaces de generar reglas para la clasificación de los datos.

Los métodos de aprendizaje basados en reglas de clasificación buscan obtener reglas o árboles de decisión que particionen un grupo de datos en clases predefinidas.

15. Árboles de Decisión, Clasificación.

Un árbol de decisión es un modelo predictivo que mapea observaciones sobre un artículo a conclusiones sobre el valor objetivo del artículo. Es uno de los enfoques de modelado predictivo utilizadas en estadísticas, minería de datos y aprendizaje automático. Los modelos de árbol, donde la variable de destino puede tomar un conjunto finito de valores se denominan árboles de clasificación. En estas estructuras de árbol, las hojas representan etiquetas de clase y las ramas representan las conjunciones de características que conducen a esas etiquetas de clase.

Este método se encuentra dentro de la clasificación supervisada⁴, son representados en forma de árbol, en donde las ramas se generan de forma recursiva representando un conjunto de decisiones hasta llegar al resultado de finalización; los cuales entre sus principales ventajas presenta sencillez, representación en forma gráfica y rapidez de clasificación.

También se conoce a árbol de decisión como un modelo de predicción utilizado en el ámbito de la inteligencia artificial. Dada una base de datos se construyen diagramas de construcciones lógicas, muy similares a los sistemas de predicción basados en reglas, que sirven para representar y categorizar una serie de condiciones que ocurren de forma sucesiva, para la resolución de un problema.

En análisis de decisión, un árbol de decisión se puede utilizar para representar visualmente y de forma explícita decisiones y toma de decisiones. En minería de datos, un árbol de decisión describe datos, pero no las decisiones; más bien el árbol de clasificación resultante puede ser un usado como entrada para la toma de decisiones, ya que éste último es el resultado predicho que define la clase a la que pertenecen los datos.

Un árbol de Clasificación está conformado por un nodo raíz, nodos intermedios y nodos terminales y de acuerdo a su clasificación, si las variables son discretas es un árbol de clasificación y si son continuas es un árbol de regresión; también puede ser clasificado por el número de ramas o hijos, por ejemplo: binarios, ternarios, n-arios. Más adelante, en nuestro trabajo vamos a revisar el funcionamiento de un árbol de clasificación con weka.

A continuación se muestra un ejemplo de un árbol de clasificación (Figura 4), con factores que determinan si se puede otorgar un crédito o no a un cliente:

⁴ Se basa en el entrenamiento, en clasificación con aprendizaje.

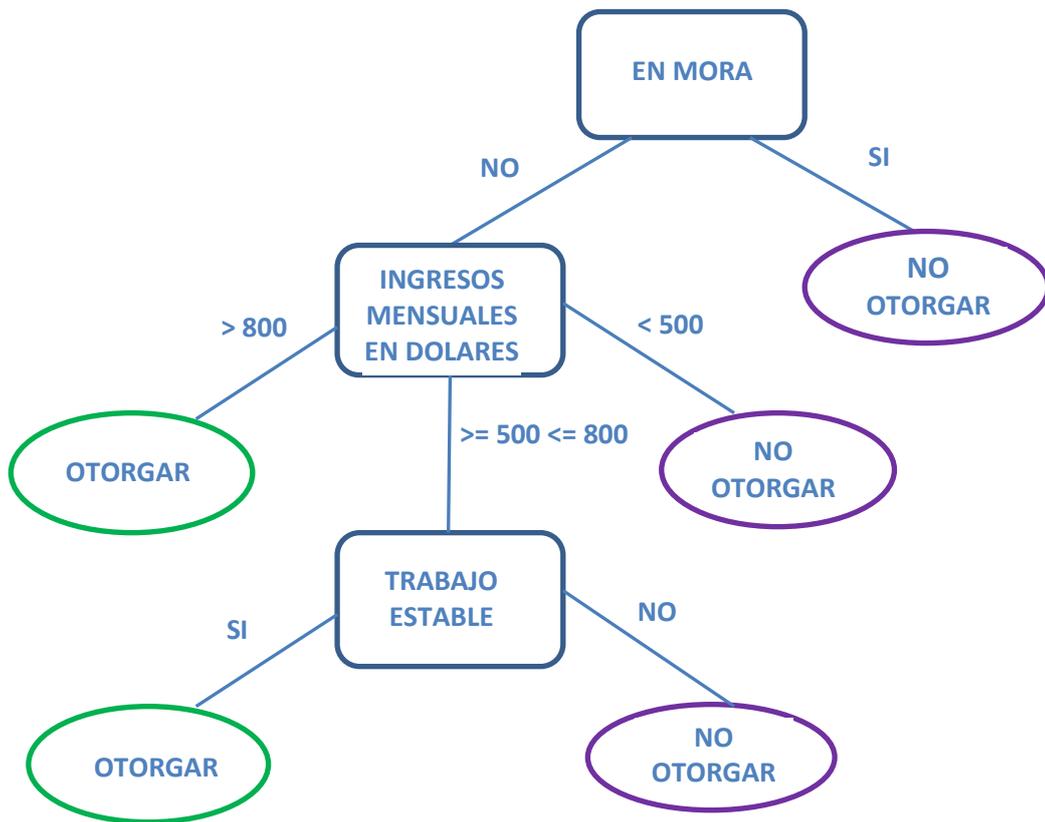


Figura 4. Árbol de clasificación.

Fuente: construcción propia.

Entre otros métodos de minería de datos, los árboles de decisión tienen varias ventajas:

- **Fácil de entender e interpretar.** Las personas son capaces de comprender los modelos de árboles de decisión después de una breve explicación.
- **Requiere poca preparación de los datos.** Otras técnicas a menudo requieren la normalización de datos, utilización de variables ficticias necesitan ser creados y valores en blanco deben ser eliminados.
- **Capaz de manejar tanto datos numéricos y categorizados.** Otras técnicas son generalmente especializadas en el análisis de conjuntos de datos que tienen sólo un tipo de variable. (Por ejemplo, las normas de relación sólo se pueden utilizar con variables nominales, mientras que las redes neuronales pueden ser utilizados sólo con variables numéricas.)
- **Utiliza un modelo de caja blanca.** Si una situación dada es observable en un modelo entonces la condición se explica fácilmente por la lógica booleana. (Un

ejemplo de un modelo de caja negra es una red neural artificial ya que la explicación de los resultados es difícil de entender.)

- **Es posible validar un modelo utilizando pruebas estadísticas.** Eso hace que sea posible tener en cuenta la fiabilidad del modelo.
- **Robusto.** Se desempeña bien incluso si sus suposiciones son violadas por el verdadero modelo a partir del cual se generaron los datos.
- **Funciona bien con grandes conjuntos de datos.** Grandes cantidades de datos pueden ser analizados utilizando recursos informáticos estándar en un plazo razonable.

Un árbol de decisión puede tener las siguientes limitaciones:

- Los algoritmos no pueden garantizar devolver el árbol de decisión globalmente óptimo.
- Aprendices de árbol de decisiones pueden crear árboles excesivamente complejos que no generalizan bien a partir de los datos de entrenamiento.

1.6. Redes neuronales artificiales.

Una red neuronal es un procedimiento de tratamiento de información que toma como ejemplo la neurona del ser vivo, siendo ésta última una unidad estructural en forma de árbol que se compone esencialmente de tres partes: el cuerpo principal o soma, el axón, y las dendritas. El cuerpo principal recibe las señales de entrada por medio de las dendritas, procesa la información, la organiza y envía los resultados de salida; aquí también se encuentra el núcleo de la neurona y la información interna. El axón tiene dos funciones primordiales: mediante un proceso llamado sinapsis se encarga transmitir información que recibe desde el cuerpo principal y comunicarse con otras neuronas por medio de sus terminales finales (Figura 5).

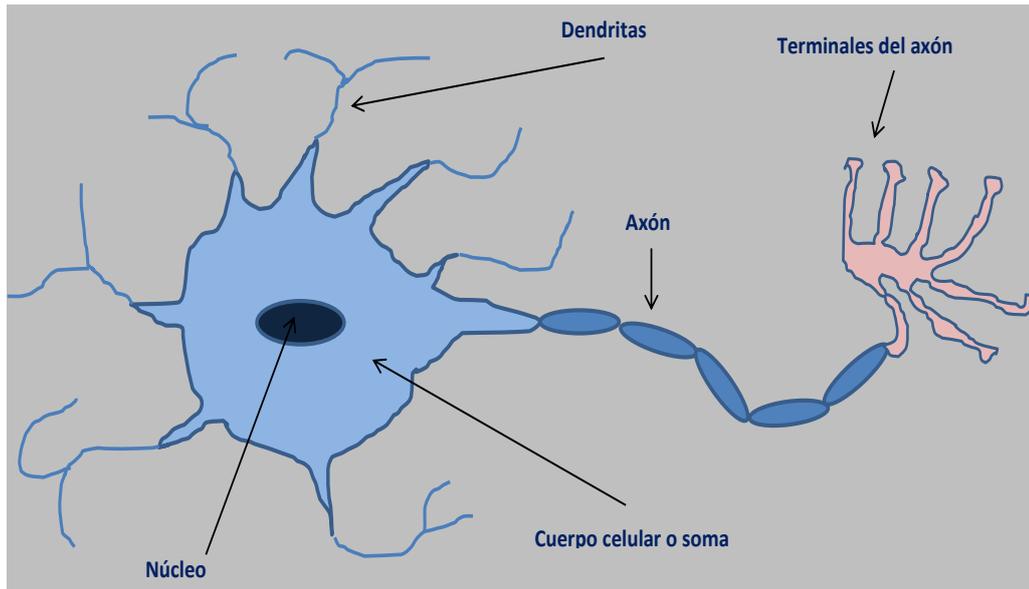


Figura 5. Neurona biológica.

Fuente: construcción propia.

Según Niesen H., las RNA son *“Un sistema de computación hecho por un gran número de elementos simples, elementos de proceso muy interconectados, los cuales procesan información por medio de su estado dinámico como respuesta a entradas externas”*, Kohonen, la define, *“Redes interconectadas masivamente en paralelo de elementos simples (usualmente adaptativos) y con organización jerárquica, las cuales intentan interactuar con los objetos del mundo real del mismo modo que lo hace el sistema nervioso biológico”*.

Para el funcionamiento de la red neuronal se recibe información de una fuente externa por medio de un estímulo, se la procesa y presenta un resultado; simulando el proceso del sistema nervioso para aprender y memorizar.

Por lo general una neurona artificial se puede representar con los siguientes elementos:

- **Entradas:** denominadas por el símbolo X, y según el número de entradas pueden ser X1, X2, X3, X4,....., Xn.
- **Pesos:** La información que ingresa dentro de la neurona es información general y sin tratamiento, la cual se le dará un peso de acuerdo a la importancia de la información, estos pesos serán representados por medio de W, así tendremos: W1, W2, W3, W4,....., Wn.
- **Regla de Propagación:** Al valor X de entrada siempre se le multiplica por el peso W por función de ponderación de información de entrada F(X):

$$F(X) = \sum^n X_i W_j$$

- **Función de Transferencia:** Una vez que la función de transferencia ha sumado toda la información y el resultado ha superado una cantidad llamada umbral (Θ)ⁿ, entra en juego la función de Transferencia o de Activación en donde recibe la suma ponderada de la función de transferencia y calcula la salida.

$$F(T) = f(\sum^n X_i W_j - \Theta_i)$$

- **Salida:** el resultado final se representa por medio de la letra Y. La salida de una red neuronal consiste en una combinación lineal de sus entradas.

Todos los elementos descritos anteriormente forman un proceso simple de cálculo de entrada, proceso de información y salida de resultados (Figura 6).

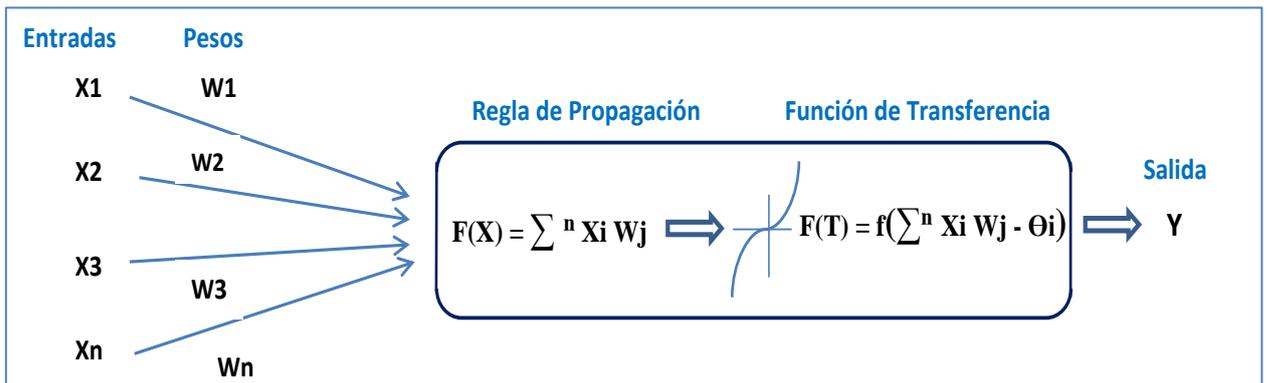


Figura 6. Procesos de una red neuronal.

Fuente: construcción propia.

Los componentes de una red neuronal son los siguientes:

Los componentes de una red neuronal se pueden clasificar en tres partes (Figura 7):

- **Capa de entrada:** en donde se recibe la información de entrada que llega desde su entorno exterior.
- **Capa oculta:** este nivel se encarga del procesamiento de la información y de generar enlace hacia otras capas. No tienen comunicación directa con el entorno exterior.
- **Capa de salida:** en esta capa se envía el resultado o la respuesta de la red neuronal.

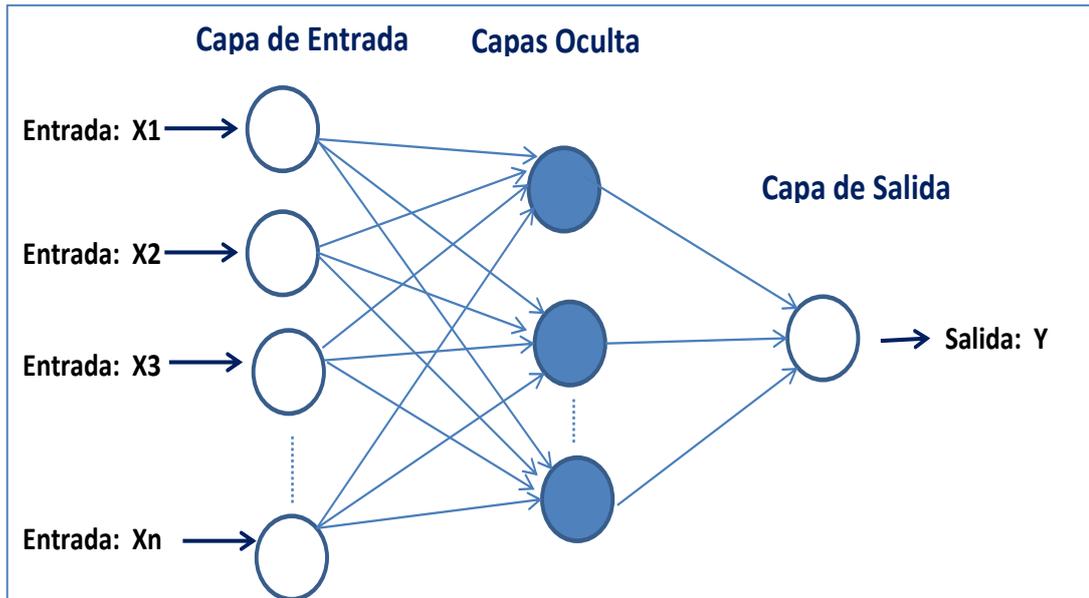


Figura 7. Componentes de una red neuronal.

Fuente: construcción propia.

1.7. El Perceptrón multicapa.

El Perceptrón⁵ multicapa es un modelo supervisado de red neuronal, el mismo que se encuentra conformado por una capa de entrada, una o varias capas ocultas y una capa de salida; comúnmente se utiliza el algoritmo o método de aprendizaje de backpropagation para la optimización de resultados.

Entre sus principales características tenemos:

- **Reconocimiento, categorización y clasificación supervisada de la información:** lo cual permite aprender de la información de entrada y salida, sobre todo en donde la clasificación de datos y predicción de resultados se hace necesario.
- **Potencia y generalidad:** ya que cumple la función de un aproximador universal de funciones (Hornik, 1989); lo que permite que con entradas que nunca han sido procesadas se genere un óptimo resultado, debido a su flexibilidad y adaptabilidad.
- **Predicción de resultados:** se basa en la obtención de una variable definida de predicción; en base al ingreso y tratamiento de varias variables de predicción ingresadas.

El Perceptrón Multicapa se compone por una capa de entrada, una o varias capas ocultas y una capa de salida (Figura 8).

⁵ En 1959, Rosenblat definió el Perceptrón, uno de los conceptos más importantes dentro del desarrollo de las Redes Neuronales

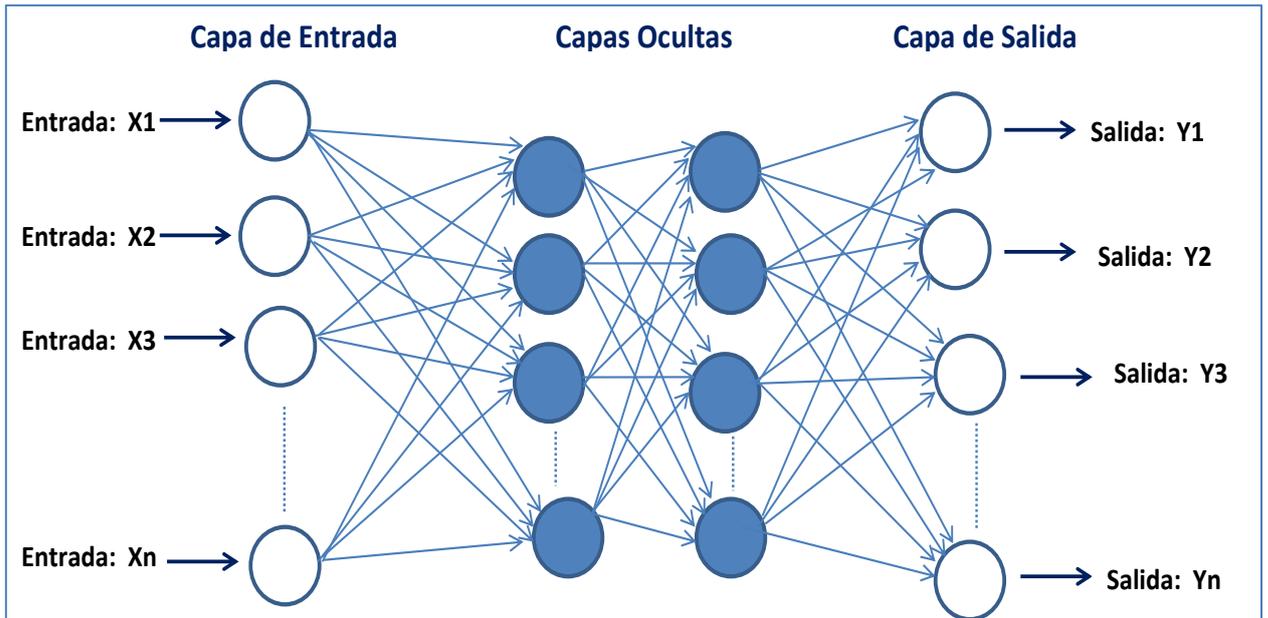


Figura 8. El Perceptrón Multicapa.

Fuente: construcción propia.

En el Perceptrón Multicapa las capas ocultas realizan la suma ponderada de las entradas con los pesos sinápticos, y al ser un modelo de aprendizaje supervisado el aprendizaje lo realizan por medio del algoritmo de backpropagation⁶.

La información dentro de ésta red siempre se transmite desde la capa de entrada hasta la capa de salida. Cuando ingresa una entrada a la red: $X_1, X_2, X_3, X_4, \dots, X_n$, se realiza una ponderación por medio de los pesos e ingresa a las capas ocultas, en donde se procesan las señales o entradas mediante la función de Activación, generando un valor de salida.

Por términos de notación vamos a definir las siguientes variables:

- Entrada neta como Ent .
- W_{ji} , como el peso de conexión entre la neurona de entrada i y la neurona oculta j ,
- Z_{kj} como el peso de conexión entre la neurona oculta j y la neurona de salida k .
- θ_j es el umbral de la neurona, con un peso asociado a un valor de salida igual a 1.
- Y_j es el valor de salida de las neuronas j de las capas ocultas hacia la capa de salida.
- Y_k es el valor resultante de las neuronas de la capa de salida hacia el exterior.

⁶ Significa propagación hacia atrás de errores, es un algoritmo supervisado para entrenar redes neuronales artificiales.

En base a lo anterior vamos a definir el siguiente procedimiento del Perceptrón, el cual funciona en base al algoritmo de backpropagation:

- La entrada total o neta que reciben las neuronas j de la capa oculta va a estar definida como:

$$Ent_j = \sum_{i=1}^N W_{ji} X_i + \theta_j$$

- El valor de salida de la neurona j , Y_j , se obtiene mediante una función sobre la entrada total:

$$Y_j = f(Ent_j)$$

- La entrada total que ingresa hacia la neurona de salida k es el siguiente resultado:

$$Ent_k = \sum_{j=1}^M Z_{kj} Y_j + \theta_j$$

- El valor resultante final Y_k de la neurona de salida k por consiguiente es:

$$Y_k = f(Ent_k)$$

Como el aprendizaje de la red es de modo supervisado mediante el algoritmo de backpropagation, el objetivo es minimizar el error entre el resultado obtenido y lo que el usuario necesita; para ello, dicho error se calcula de la siguiente manera:

$$Error^p = \frac{1}{2} \sum_{k=1}^M (R_k - Y_k)^2$$

En donde R es la salida deseada para la neurona de salida k , de acuerdo a los patrones establecidos p y definir la comparación. En base a esto se puede tener una sumatoria total del error:

$$Error = \sum_{p=1}^P Error^2$$

Después de tener el valor total del error, por medio del algoritmo de backpropagación se realiza una comparación de los errores hacia atrás y modificación de los pesos hasta ajustarlos y acercarlos al valor necesitado; por ejemplo un valor de una capa oculta viene determinado por la suma de los errores que se produjeron en las neuronas de salida k en las cuales se procesaron con anterioridad las entradas de las neuronas j de la capa oculta y el

proceso seguirá hasta llegar al objetivo final; por lo tanto a éste proceso también se lo conoce como propagación del error hacia atrás.

1.8. Ventajas y desventajas del uso de la IA.

En el ámbito mundial la IA ha tomado mucha importancia en los últimos años, sobre todo en lo que al avance de la tecnología se refiere, no solo de ayuda para eventos científicos sino también para la sociedad en general, y entonces podemos definir a la IA en este entorno como "*La ciencia que enfoca su estudio a lograr la comprensión de entidades inteligentes. Es evidente que las computadoras que posean una inteligencia a nivel humano, o superior, tendrán repercusiones muy importantes en nuestra vida diaria*" (Zaccagnini y Caballero. 1992:24).

La utilización de la IA en la actualidad está basada principalmente en optimizar y acelerar la productividad del ser humano, constituyendo de ésta forma un avance tecnológico; pudiendo realizar tareas como pasear un perro, traer la correspondencia o proteger una gran corporación. Entre sus ventajas podemos citar las siguientes, tomando en cuenta un elemento tecnológico como por ejemplo un sistema experto⁷:

- Puede tener más memoria que un ser humano
- Pensar lógicamente más rápido que un ser humano
- Puede crear relaciones indefinidamente
- Agilizar cálculos complejos para el ser humano y presentar cálculos exactos
- Ayuda para acelerar tareas mentales y físicas del ser humanos, por ejemplo un robot computarizado (Figura 9).

Entre sus desventajas pueden estar:

- Falta de capacidad de pensamiento crítico independiente como un ser humano.
- No tiene sentido común, ni sentimientos.
- No tiene iniciativa propia ni creatividad.

⁷ Es una aplicación informática capaz de solucionar un conjunto de problemas que exigen un gran conocimiento sobre un determinado tema

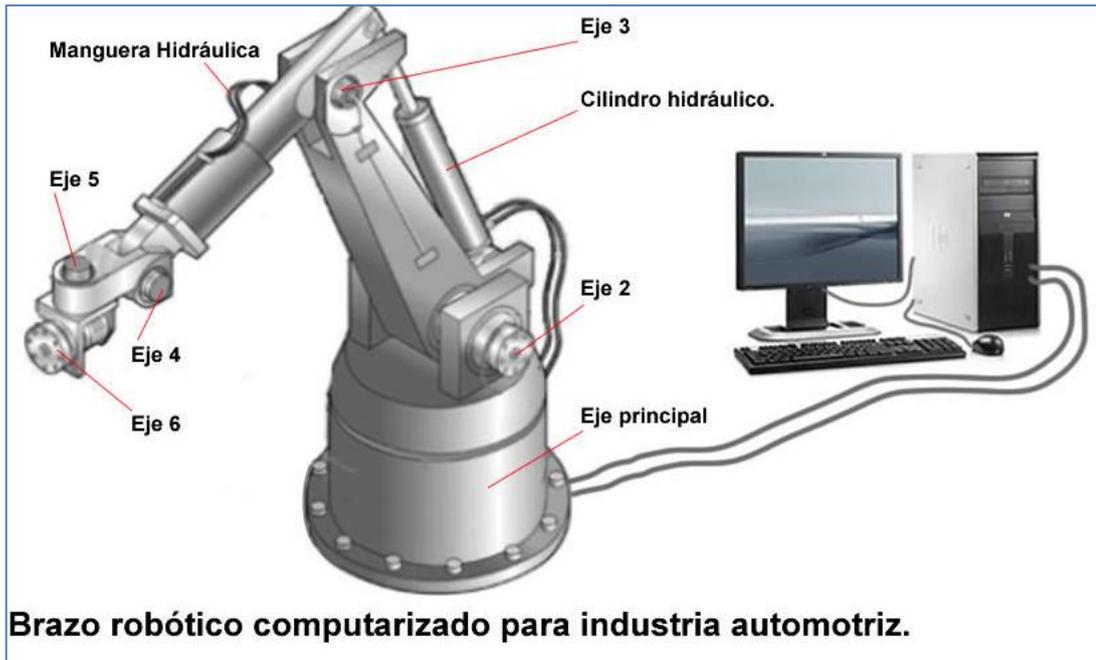


Figura 9. Robot computarizado.

Fuente: <http://www.inventosmodernos.cl/robot.htm>

CAPITULO II
LA INTELIGENCIA ARTIFICIAL APLICADA AL RIESGO CREDITICIO

2.1. Concepto de riesgo crediticio.

El crédito se puede definir como la confianza que posee una persona para poder entregarle un servicio, en nuestro caso éste va a ser un préstamo; el cual cumple con un proceso o ciclo (Figura 10). Se denomina riesgo crediticio a la posible eventualidad de que una persona natural o jurídica no cumpla a tiempo con las obligaciones acordadas con la entidad financiera en base al crédito, es decir, que no cancele el dinero del préstamo en el tiempo acordado, con lo cual es una puerta de entrada a problemas financieros.

Toda institución financiera puede incurrir hoy en día de riesgos de crédito si no se toman las debidas precauciones, para poder entregar un crédito a un cliente se debe analizar correctamente las variables y resultados previos obtenidos, y si existieron errores se puede aprender de ellos. Un riesgo de crédito se entiende como el derivado de cambios en la clasificación crediticia del emisor generado por la probabilidad de incurrir en pérdidas por el impago en tiempo o forma de las obligaciones crediticias de uno o varios prestatarios (de Lara, 2009).

El riesgo crediticio se encuentra ligado a varios factores que se toman en cuenta al momento de evaluar al cliente, tales como capacidad financiera, estado civil, tipo de trabajo, edad, etc.; para los cual se hace necesario tener un buen personal evaluador dentro de la institución financiera y sobre todo con buenas herramientas, técnicas y metodologías que ayuden a evitar que la empresa tenga futuros problemas de tipo financiero y que sus clientes no caigan en la morosidad y central de riesgos.



Figura 10. Ciclo del crédito.

Fuente: construcción propia

2.2. Factibilidad de aplicación en procesos financieros.

Se ha demostrado que los procedimientos de la Inteligencia Artificial son bastante eficientes para predecir variables financieras, sobre todo aprovechando la característica de aprender que éstas poseen. En base a esto se considera a una red de cualquier modelo que sea como en equipo de cómputo que procesa entradas y presenta resultados.

Como el tema principal de éste trabajo tiene que ver el riesgo crediticio por motivo de préstamos a clientes dentro de una institución financiera, vamos a elaborar un procedimiento que sea de gran ayuda para verificar los riesgos que se pueden presentar; con los resultados obtenidos podemos verificar los logros en comparación con los métodos tradicionales.

A nivel internacional se creó el acuerdo de Basilea⁸ en 1988 en el cual se sugiere que las entidades financieras modernas deben utilizar metodologías modernas y tecnología para afrontar o problemas en las instituciones financieras, entre ellos el riesgo crediticio, en predicción de la quiebra y en la estimación de los rendimientos de cartera de valores.

Existen en el medio informático, herramientas de IA que son gratuitas⁹, no necesitan muchas exigencias de hardware y software, además son factibles de utilizar en los equipos computacionales de la institución financiera del objeto de nuestro estudio y nos van a servir para realizar la clasificación de la información y predicción de resultados, en el capítulo siguiente vamos a estudiar el funcionamiento de dicha herramienta. Otro aspecto de gran ayuda es el internet, en donde podemos encontrar ayuda e información sobre los aspectos de IA, ya que existe amplia publicación acerca de éste tema.

2.3. Estudio del caso: entidad financiera en donde se aplicará el modelo de gestión de riesgos.

El presente trabajo de investigación se realizará en la Cooperativa de Ahorro y Crédito Nuevo Amanecer, CACNA, ubicada en Llano Grande por el sector norte de la ciudad de Quito; en donde se otorgan aperturas de cuentas de ahorros para los ciudadanos y microcréditos para comerciantes y personas de dependencia laboral (empleados), éstos últimos son aprobados en un máximo de 72 horas si el cliente reúne los requisitos y los documentos exigidos. La entidad inició sus labores en el año 2012 y cuenta con alrededor

⁸ Acuerdos de Basilea I, Basilea II y Basilea, su nombre se debe que se celebró en dicha ciudad suiza.

⁹ Existe en el internet software de libre distribución, el cual lo podemos bajar, instalar y sin restricciones de uso.

de 150 clientes, entre ahorristas y clientes con créditos, siendo su función la de dedicarse al microcrédito, al cual la Conferencia Internacional sobre Microcrédito (97), lo define como “Programas de concesión de pequeños créditos a los más necesitados de entre los pobres para que éstos puedan poner en marcha pequeños negocios que generen ingresos con los que mejorar su nivel de vida y el de sus familias”.

A pesar de que las microfinanzas han tenido una expansión en los últimos años en el Ecuador, estas no han tenido un avance muy significativo debido al desconocimiento sobre las entidades principales que otorgan estos servicios o debido a su difícil acceso, por eso las personas tienden a utilizar servicios de préstamos informales con consecuencias negativas.

Por otro lado una empresa financiera puede tener problema de morosidad por las siguientes razones:

- Falta de información al cliente
- Al momento de otorgar un crédito el empleado muchas de las veces no da la orientación debida.
- Por lo general los empleados tratan de cumplir sus objetivos personales y no informan al clientes sobre la naturaleza, normas y procedimientos de cobranzas.
- empleados mal capacitados.
- Entrega de tarjetas de crédito con desinformación al cliente.
- Falta de procedimientos para evaluar el perfil de los clientes y determinar si son sujetos de crédito o no.

En el Anexo 1 se muestra las fotos del local de la cooperativa y su ubicación dentro del distrito metropolitano de Quito.

Para la realización de nuestras labores en la cooperativa CACNA, se cuenta con la colaboración del personal administrativo y operacional, teniendo acceso a la base de datos de la cartera de clientes de la entidad para poder realizar nuestro estudio sin inconvenientes; para lo cual inicialmente se tuvo una reunión previa con el personal explicar el fin de nuestro trabajo.

Dentro de la cooperativa se trabaja con métodos tradicionales de evaluación del cliente que quiere aplicar un préstamo, y muchas veces se realiza esta tarea tomando en cuenta la experiencia u otros factores. Con nuestro trabajo se ayudará con una herramienta que ayude a prevenir el riesgo crediticio y evaluar de mejor manera al cliente.

Con respecto a la instalación al software computacional, la plataforma escogida funcionó sin ningún problema en el computador en donde labora el oficial de crédito y en donde se va a implementar nuestro modelo.

2.4. Variables a evaluar en el riesgo crediticio.

"En el proceso de evaluación de un crédito para una empresa se debe contemplar una evaluación profunda tanto en sus aspectos cuantitativos como cualitativos" (Ellerger, 2002, p. 35).

Al momento que el cliente se acerca a la institución financiera, se le hace llenar un formulario de solicitud de crédito (Anexo 2), en donde debe ingresar un conjunto de datos para comenzar a analizar su petición; en base a estos datos el oficial de crédito tiene que determinar si el cliente aplica para el crédito, ayudándose también de otras fuentes externas como las páginas de internet para verificar el estado crediticio actual del interesado¹⁰. En cuanto a las variables que se utilizan para evaluar dentro de las entidades, dependiendo del criterio de cada una, casi la mayoría son las mismas, tales como: ingresos, cargas familiares, edad, tipo de bienes, instrucción, tipo de socio, experiencia financiera, tipo de vivienda, teléfono, vehículo, etc.; la idea de nuestro trabajo es buscar que variables son las más relevantes en el aspecto de que nos permitan evaluar el riesgo de crédito. En la siguiente tabla se muestra el listado de todas las variables crediticias (Tabla 1) que se utilizan en la cooperativa de ahorro y crédito CACNA, con la siguiente valoración: alto (1), medio (2), bajo(3), muy bajo(4), demasiado bajo(mayor que 4):

Tabla 1. Variables crediticias.

VARIABLES	VALORACION	
INGRESOS	1	> 1000
	2	>= 500 <= 1000
	3	>= 350 < 500
ACTIVIDAD	1	Comerciantes
	2	Dependencia Laboral (empleados)
DEPENDIENTES	1	No tiene
	2	>= 1 <= 3

¹⁰ Para verificar si el cliente se encuentra en mora, a esta información tienen acceso las entidades financieras.

	3	> 3 <= 6
	4	> 6
EDAD	1	> 35 <= 45
	2	> 45 <= 65
	3	< 25 <= 35
	4	> = 18 <= 25
MONTO	1	>= 1000 <= 2000
	2	> 2000 <= 3000
	3	> 3000
BENEFICIOS	1	IESS
	2	Seguro de vida
	3	Fondo Mortuorio
TELEFONO	1	Celular
	2	Fax
	3	Normal
INSTRUCCION	1	Masterado
	2	Superior
	3	Técnico
	4	Estudiante
	5	Secundaria
	6	Primaria
	7	Ninguna
ESTADO CIVIL	1	Soltero
	2	Casado
	4	Viudo
	6	Unión Libre
	7	Divorciado
TIPO DE VIVIENDA	1	Propia
	2	Compartida
	3	Arrendada
	4	Hipotecada
VEHICULO	1	Si
	2	No
SEXO	1	Masculino
	1	Femenino
TIPO DE SOCIO	1	Socio
	2	Normal
	3	Cliente Externo (Garante)

EXPERIENCIA FINANCIERA	1	Si
	2	Ninguna
MORA	N/A	Si
	N/A	No

Fuente: construcción propia.

2.5. Resultados esperados aplicando IA.

Para realizar ésta tarea tenemos en primer lugar que establecer todas las variables financiera que se van a procesar y después aplicar la metodología seleccionada que más nos acerque a la realidad; para ellos se trabajará con toda la información proporcionada de la cartera de clientes de la entidad, se realizará un data mining apoyándonos de la Inteligencia Artificial con alguna herramienta informática que más sea factible de utilizar y sea acorde con la realidad. El resultado final se hará conocer al personal que opera en el área de los créditos para la verificación respectiva y determinando las ventajas que se puedan dar en función de ayuda sobre el método tradicional; para esto se realizará una comparación de resultados al evaluar los clientes de forma tradicional versus nuestro modelo, por ejemplo si el oficial de crédito decide otorgar a determinado cliente un crédito, puede también verificar resultados de nuestro trabajo, ya que el mismo se ha hecho en base a todos los resultados anteriores en base al aprendizaje.

CAPITULO III

PLATAFORMA A UTILIZAR PARA EL TRATAMIENTO DE DATOS: WEKA.

3.1. Introducción a Weka.

Para poder implementar nuestro modelo predictivo para la detección y prevención de riesgo crediticio se va a utilizar la plataforma de software Weka, con el cual se puede extraer conocimientos de grandes bases de datos y trabajar con redes neuronales aprovechando las bondades de éste software para el aprendizaje automático; se puede decir también que es un programa ampliamente utilizado y probado dentro del análisis predictivo y minería de datos, posee herramientas sencillas y potentes, lo cual nos será de gran ayuda para la consecución de nuestro trabajo ya que se realizarán tareas de clasificación y predicción. La pantalla principal de Weka (Figura 11) es la siguiente:



Figura 11. Pantalla principal de Weka.

Fuente: resultado de ejecución de Weka.

Weka quiere decir: Waikato Environment for Knowledge Analysis, en español: entorno para análisis del conocimiento de la Universidad de Waikato¹¹.

3.2. Ventajas de utilizar Weka.

Entre sus ventajas de Weka podemos anotar:

- Es un software libre distribuido bajo la licencia GNU-GPL y no se nos dificulta poder conseguirlo e instalarlo.
- Puede correr en muchas plataformas debido a que está completamente desarrollado en Java. Se comprobó que funciona sin problemas para Windows 7, el sistema operativo que en nuestro caso utilizamos.

¹¹ Weka fue desarrollado en la universidad de Waikato en Nueva Zelanda en el año de 1993, y se han realizado mejoras e implementaciones desde sus inicios.

- Contiene muchos algoritmos de clasificación, así como técnicas de procesamiento y modelado de datos.
- Es relativamente fácil de utilizar debido a su interfaz gráfica de usuario.

Una de las desventajas del programa Weka es que no hay mucha documentación para guía del usuario; aunque últimamente se puede encontrar información relevante en el internet.

En los siguientes capítulos vamos a ir verificando el funcionamiento de la plataforma Weka en conjunto con la información recopilada dentro de cooperativa Nuevo Amanecer y en base a los resultados deseados del riesgo de crédito.

3.3. Gestión de datos y tipos de variables.

Los datos de entrada para la herramienta Weka deben estar codificados en un formato específico denominado arff, lo cual quiere decir *Attribute-Relation File Format*. Con dicho formato se genera un archivo sobre el que se van a implementar las técnicas de nuestro interés y se encuentra conformado por tres partes:

- **Cabecera:** @relation <nombre-de-la-relación>
- **Declaración de atributos:** @attribute <nombre-del-atributo> <tipo>
- **Sección de Datos:** Se representa de la siguiente forma:

@data

José, Sano, 43, masculino

De acuerdo a lo anterior vamos a ver una representación simple de un archivo llamado *salud.arff* del estado de salud de una persona, su edad y sexo:

@relation salud

@attribute nombre STRING

@attribute estado {sano, enfermo, grave}

@attribute edad INTEGER

@attribute sexo {masculino, femenino}

@data

Alberto,sano,43,masculino

Susana,enfermo,50,femenino

Roberto,enfermo,60,masculino

Eduardo,sano,22,masculino

Dicho archivo primeramente lo elaboramos en un editor de texto o Word y después lo podemos convertir al formato de weka, el cual por default es *.arff* y es de *formato compatible* para trabajar en weka, pero también podemos trabajar sin problemas con ficheros tipo *.csv*. En el Anexo 3 mostraremos en detalla los componentes del archivo tipo *.arff* y como crearlo.

3.4. Elaboración del archivo de datos para procesamiento.

Una vez que ya sabemos cómo elaborar la estructura del archivo para trabajar dentro de weka, primeramente a darle forma a nuestro fichero real llamado *riesgo.credito.arff*, el mismo que contiene todas las variables utilizadas dentro de la cooperativa para la detección del riesgo crediticio, las mismas que ya fueron definidas en el capítulo anterior. El nombre de nuestra relación se va a llamar *riesgo.credito* y va a quedar estructurada de la siguiente manera en base a las variables y pesos tomados de la tabla:

```
@relation riesgo.credito

@attribute sexo {M, F}
@attribute edad INTEGER
@attribute ingresos REAL
@attribute dependientes INTEGER
@attribute estado.civil {casado, soltero, viudo, union_libre, divorciado}
@attribute instruccion{masterado, superior, tecnico, estudiante, secundaria, primaria, ninguna}
@attribute actividad {primaria, secundaria}
@attribute vivienda {propia, compartida, arrendada, hipotecada}
@attribute monto REAL
@attribute beneficios {IESS, seguro_de_vida, fondo_mortuario}
@attribute telefono {celular, fax, normal}
@attribute vehiculo {si, no}
@attribute tipo.cliente {socio, normal, externo}
@attribute experiencia.financiera {si, no}
@attribute mora {si, no}

@data
M,41,1100,3,casado,secundaria,primaria,hipotecada,3000,IESS,celular,si,socio,si,no
.....
```

En el Anexo 4 se presenta el detalle del archivo *riesgo.credito*, en su totalidad.

CAPITULO IV
PROCESAMIENTO DE LA INFORMACION EN WEKA.

4.1. Ejecución e ingreso al programa.

Una vez que tenemos elaborado nuestro archivo de datos para ser procesado en weka, ya sea en formato *.arff*, procedemos al procesamiento de la información la cual será detallada en los pasos siguientes. En el Anexo 5 presentando un detalle más amplio de cada uno de los componentes de weka utilizados en el presente trabajo.

Nuestro principal objetivo con éste capítulo es clasificar a los clientes mediante las variables o sus atributos y verificar los factores por el cual se les pudo otorgar o no un crédito.

Al ejecutar weka nos presenta la siguiente pantalla principal con los siguientes componentes o módulos (Figura 12), vamos a ir utilizando el programa con nuestros datos a procesar:



Figura 12. Módulo de Weka.

Fuente: resultado de ejecución de Weka.

Esta primera pantalla nos muestra una serie de módulos para utilizar en nuestro proyecto¹², la que más nos interesa y contiene los subprogramas de Weka es la sección de Applications, la cual se muestra en la figura anterior. El más importante es Explorer, la que nos va a permitir explorar y procesar los datos y es el que nos concierne de acuerdo a nuestro tema de estudio.

¹² Para utilizar en nuestro proyecto solamente requerimos utilizar el módulo Explorer.

4.2. Módulo Explorer.

Cuando ingresamos al Módulo Explorer (Figura 13) verificamos que éste a su vez se encuentra dividido en las siguientes secciones: Preprocess, Classify, Cluster, Associate, Select Attributes y Visualize:

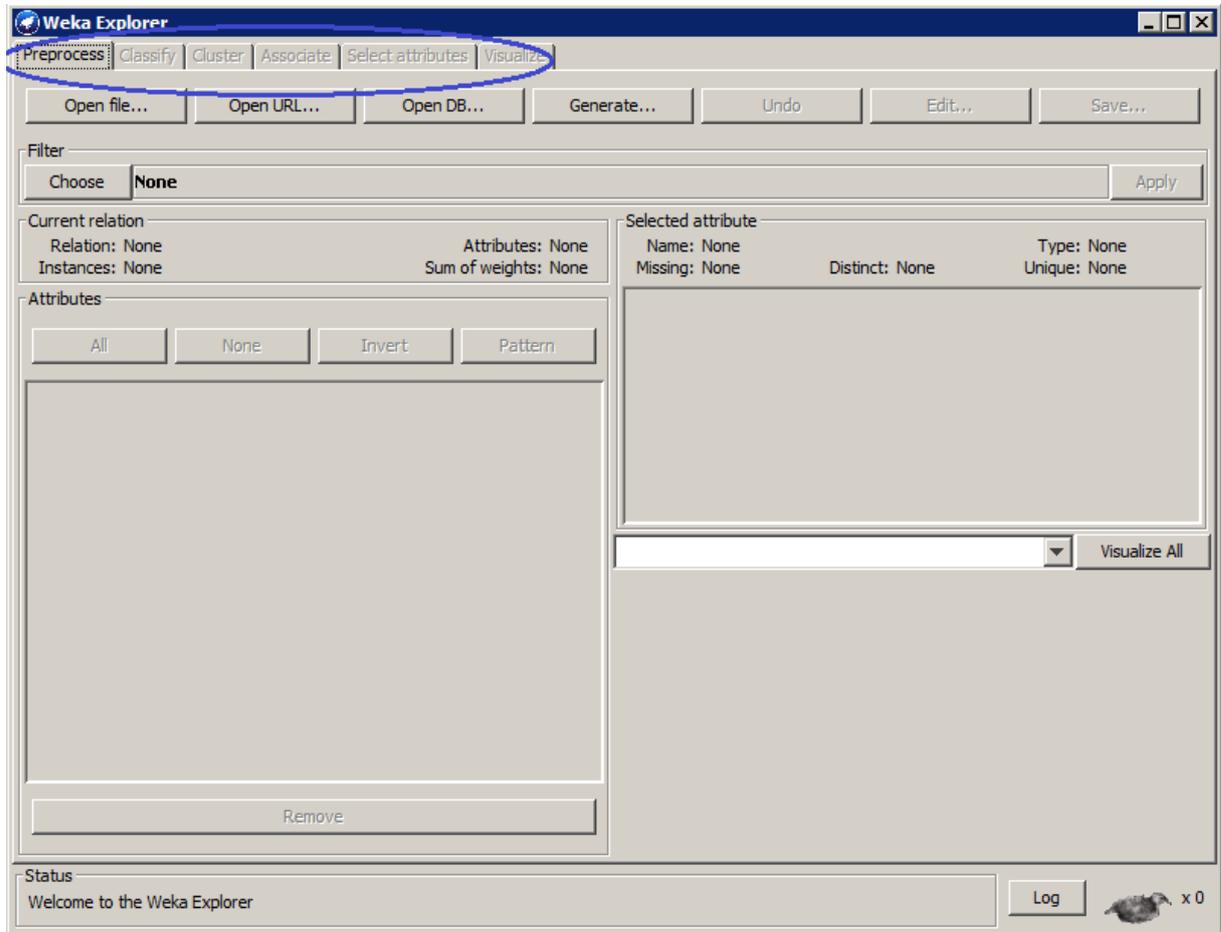


Figura 13. Módulo Explorer.

Fuente: resultado de ejecución de Explorer.

Dentro del Explorer es donde vamos a utilizar nuestro archivo *riesgo.credito*, el cual verificamos que no existe ningún error de escritura para que pueda ser utilizado sin ningún problema en el programa y seguidamente vamos a revisar el funcionamiento de las funciones o sub módulos de Explorer.

4.3. Componentes principales de Explorer.

4.3.1. Funcionamiento de Preprocess.

La función Preprocess (Figura 14) tiene que ver con el preprocesamiento de la información, en el cual podemos verificar y escoger los distintos algoritmos de aprendizaje para poder aplicarlos en el conjunto de datos.

En la opción Open File cargamos nuestro archivo y nos va a aparecer la siguiente pantalla:

The screenshot shows the Weka Explorer interface in the Preprocess mode. The main window title is 'Weka Explorer'. Below the title bar, there are tabs for 'Preprocess', 'Classify', 'Cluster', 'Associate', 'Select attributes', and 'Visualize'. The 'Preprocess' tab is active. At the top, there are buttons for 'Open file...', 'Open URL...', 'Open DB...', 'Generate...', 'Undo', 'Edit...', and 'Save...'. Below these is a 'Filter' section with a 'Choose' button and a dropdown set to 'None', and an 'Apply' button. The main area is divided into two panes. The left pane, titled 'Current relation', shows 'Relation: riesgo.credito' and 'Instances: 48'. Below this is a list of 'Attributes' with buttons for 'All', 'None', 'Invert', and 'Pattern'. A table lists 15 attributes, with 'mora' selected at the bottom. The right pane, titled 'Selected attribute', shows 'Name: mora', 'Missing: 0 (0%)', 'Distinct: 2', and 'Type: Nominal'. Below this is a table with columns 'No.', 'Label', and 'Count', showing '1 si' with count 20 and '2 no' with count 28. At the bottom of the right pane is a bar chart with two bars: a blue bar for 'si' (count 20) and a red bar for 'no' (count 28). The status bar at the bottom shows 'Status OK' and a 'Log' button.

Figura 14. Función Preprocess.

Fuente: resultado de ejecución de Preprocess.

Una vez que hemos ingresado nuestra base de datos, la pantalla principal nos muestra en la parte izquierda la información de nuestra base, tales como Relation (nombre de la relación: *riesgo.credito*), número de atributos (16), número de instancias (48). En la parte derecha podemos ver la clasificación estadística de los atributos (Tabla 2), y en base a esto hemos generado la siguiente tabla:

Tabla 2. Clasificación estadística de los atributos

ATRIBUTO	VALOR	CANTIDAD	%	MINIMO	MAXIMO	MEDIA	DESV. ESTÁNDAR
SEXO	M	21	43.75%				
	F	27	56.25%				
EDAD				20	77	43.354	14.69
INGRESOS				350	1300	688.542	257.112
DEPENDIENTES				0	7	2.958	1.663
ESTADO CIVIL	casado	37	75				
	soltero	5	10.41%				
	viudo	1	2.08%				
	union_libre	3	6.25%				
	divorciado	3	6.25%				
INSTRUCCIÓN	masterado	0	0\$				
	superior	4	8.33%				
	técnico	2	4.16%				
	estudiante	11	22.92\$				
	secundaria	26	54.16\$				
	primaria	5	10.41\$				
	ninguna	0	0\$				
ACTIVIDAD	primaria	18	37.50%				
	secundaria	30	62.50%				
VIVIENDA	propia	17	35.42%				
	compartida	4	8.33%				
	arrendada	20	41.66%				
	hipotecada	7	14.58%				
MONTO				300	5000	1214.69	991.818
BENEFICIOS	IESS	48	100%				
	seguro de vida	0	0%				
	fondo mortuario	0	0%				
TELEFONO	celular	46	95.83%				
	fax	0	0\$				
	normal	2	4.16%				
VEHICULO	si	26	54.16%				
	no	22	45.83%				
TIPO CLIENTE	socio	1	2.08%				
	normal	45	93.75%				
	externo	2	4.16%				
EXPERIENCIA FINANCIERA	si	30	62.50%				
	no	18	37.50%				
MORA	si	20	41.67%				
	no	28	58.33%				

Fuente: construcción propia.

Para poder obtener los valores seleccionamos previamente el atributo deseado, por ejemplo *edad*, de donde tomamos valor mínimo, valor, máximo, media aritmética (Mean) y desviación estándar (StdDev), para obtener valores estadísticos (Figura 15) tal como se muestra en la siguiente figura:

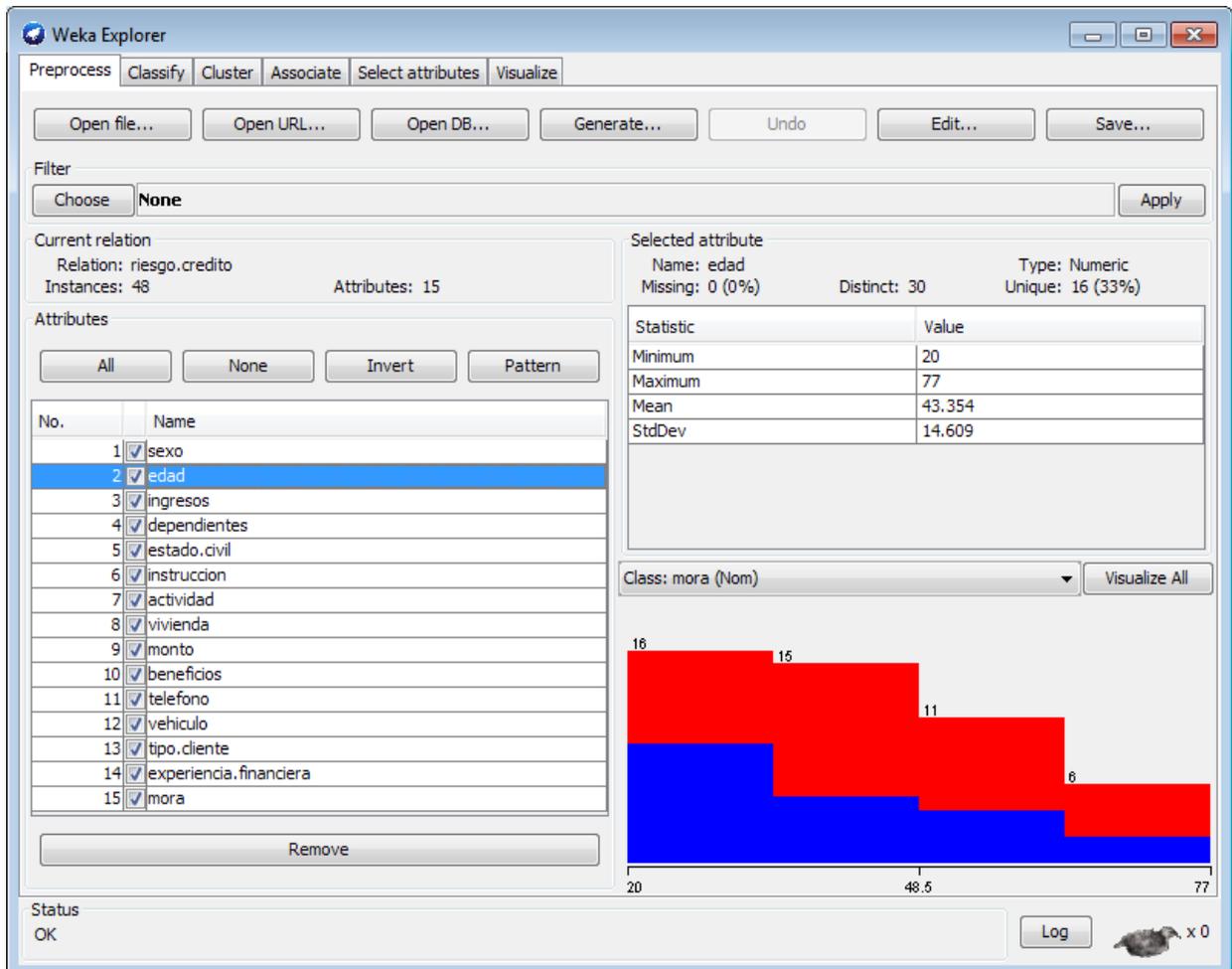


Figura 15. Valores estadísticos.

Fuente: resultado de ejecución de Preprocess.

El valor *Unique* se refiere al número de valores que solo aparecen una vez en el atributo y *Distinct* tiene que ver al número de valores distintos, no tomando en cuenta los repetidos.

En el gráfico de la parte inferior derecha aparecen los valores de los atributos divididos por su clase (Figura 16), en el eje X nos detalla los valores del atributo y en el eje Y la participación de las instancias en el rango de los valores de los atributos. Por ejemplo, en el grafico siguiente, tomamos como ejemplo el atributo ingresos, en donde:

- 18 personas perciben un sueldo de 354 \$ hasta 590.5 \$.
- 17 personas tienen un sueldo de 590.5 \$ hasta 827 \$.

- 6 personas tiene un sueldo desde 827 \$ hasta 1063.5 \$.
- 7 personas perciben un ingreso desde 1063.5 \$ hasta 1300 \$

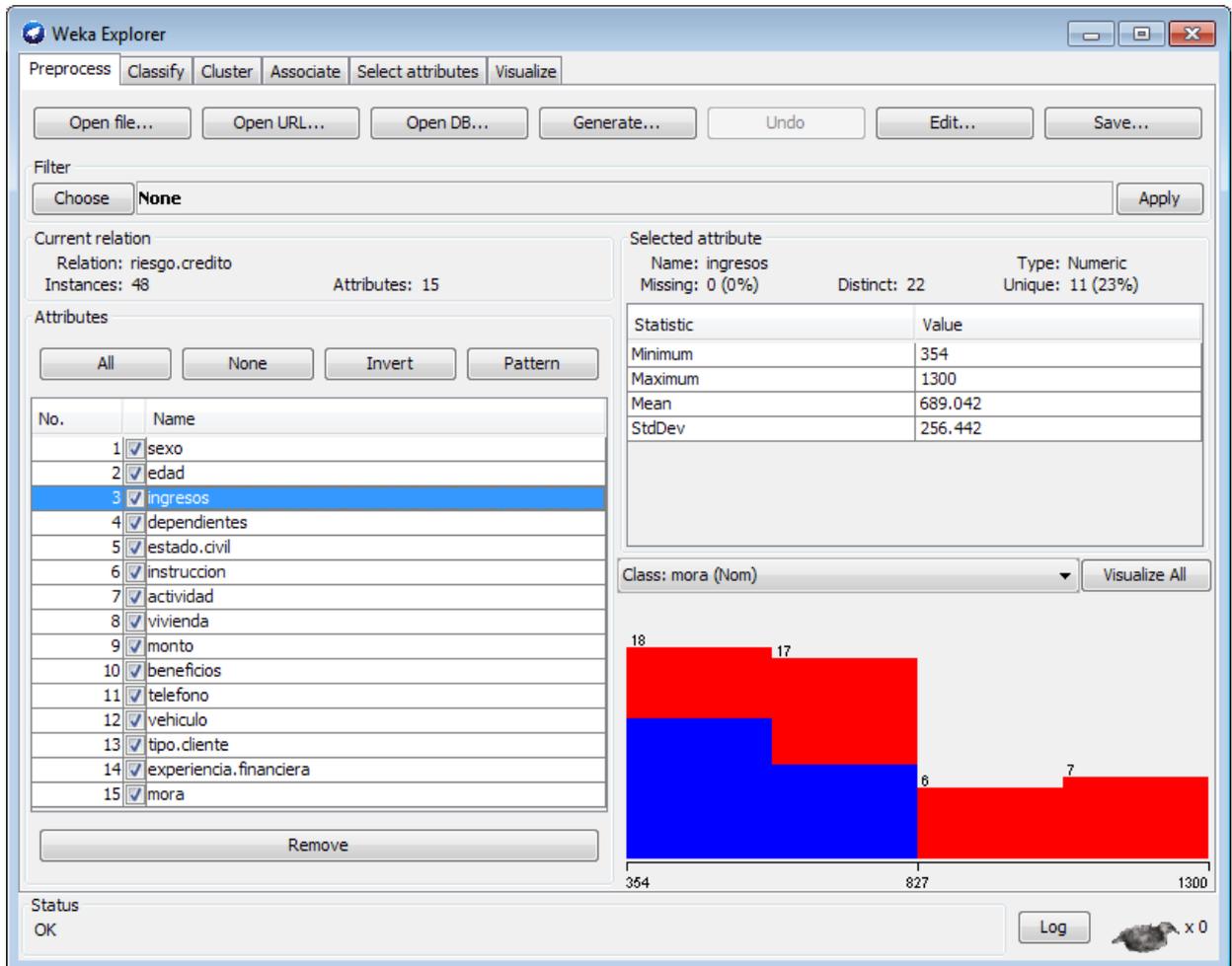


Figura 16. Atributos divididos por su clase.

Fuente: resultado de ejecución de Preprocess.

Si queremos revisar los detalles de los atributos (Figura 17) que se utilizan en la gestión de créditos de los clientes en la cooperativa, pulsamos la opción Visualize All:

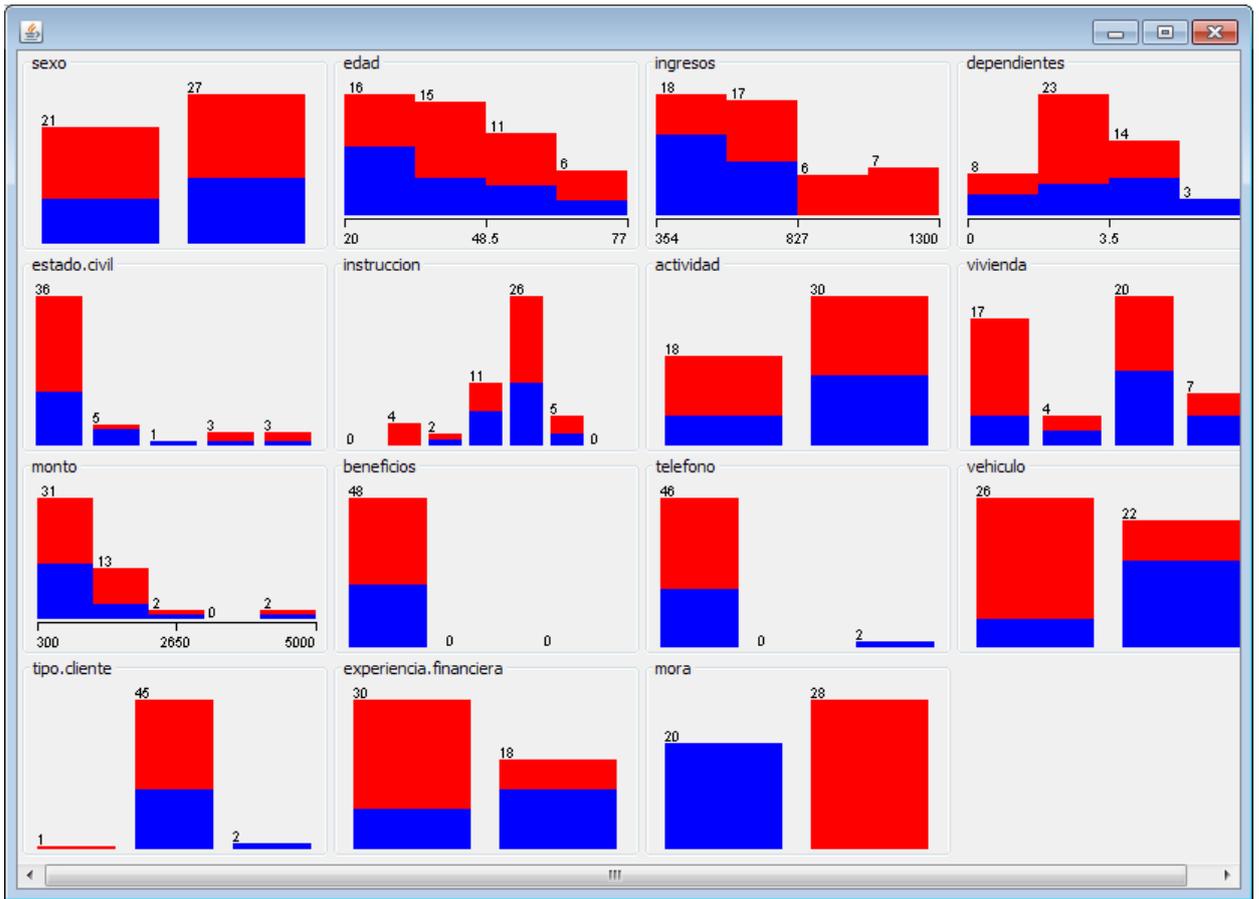


Figura 17. Detalles de los atributos.

Fuente: resultado de ejecución de Preprocess.

4.3.2. Funcionamiento de Classify.

La función Classify (Figura 18) dentro de Weka nos sirve para realizar la clasificación de los datos que poseemos. Debajo tenemos las opciones de test options, en donde vamos a procesar el porcentaje esperado de aciertos, entre estas opciones tenemos: Use training set, Supplied test set, Cross-validation y Percentage Split.

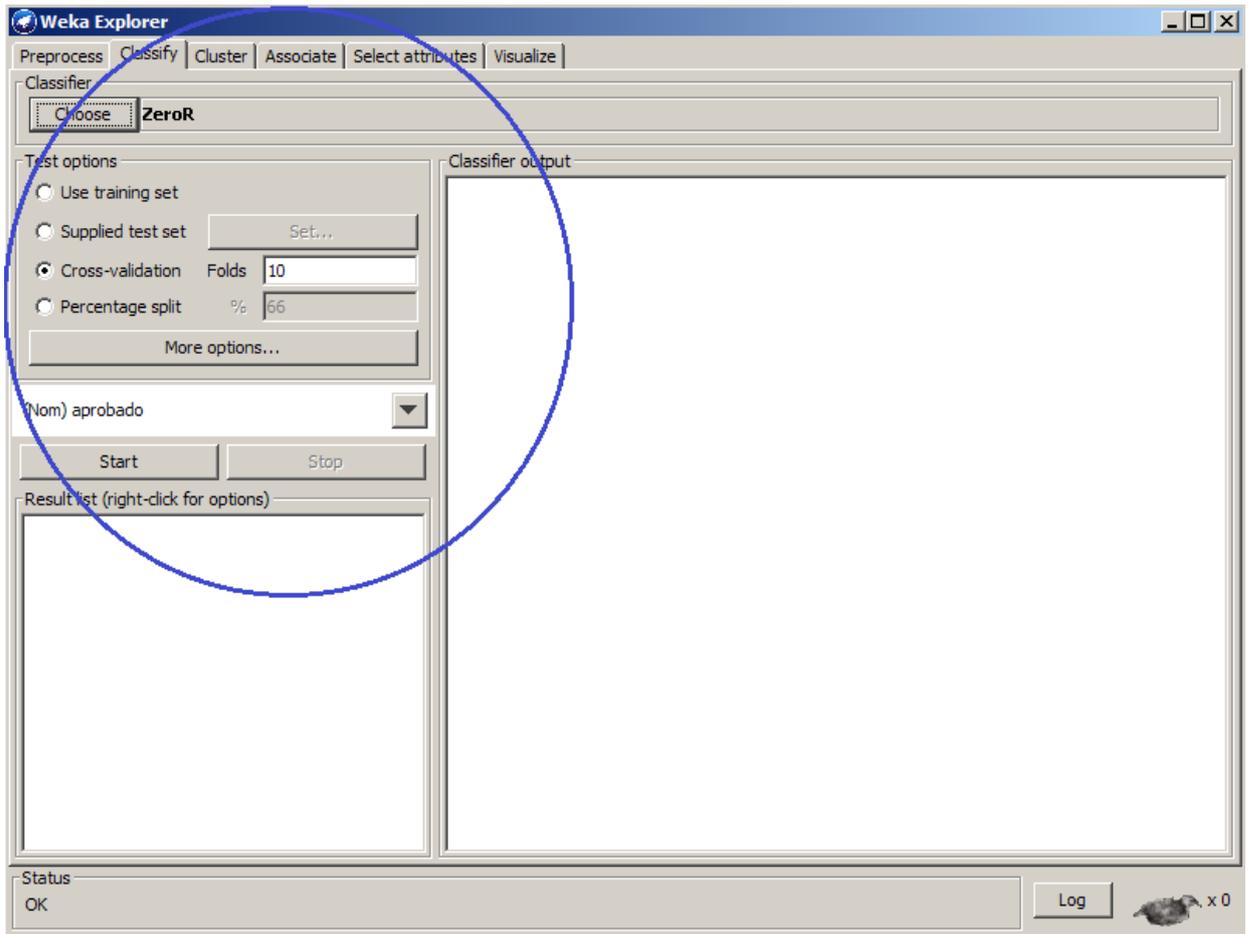


Figura 18. Función Classify.

Fuente: resultado de ejecución de Classify.

Para nuestro proyecto vamos a utilizar un Clasificador con el cross-validation para el proceso de los datos; en primer término vamos a usar el algoritmo ZeroR pulsando Start con un folder de 10 que viene por default, lo cual podemos ver que no presenta resultados del clasificador (Figura 19) muy alentadores, ya que arroja un total de 28 aciertos frente a 20 errores; en porcentajes equivale a 58.33 % de aciertos y 41.66 % de errores; también se comprobó con otros números de folders, pero igual nos dio el mismo resultado.

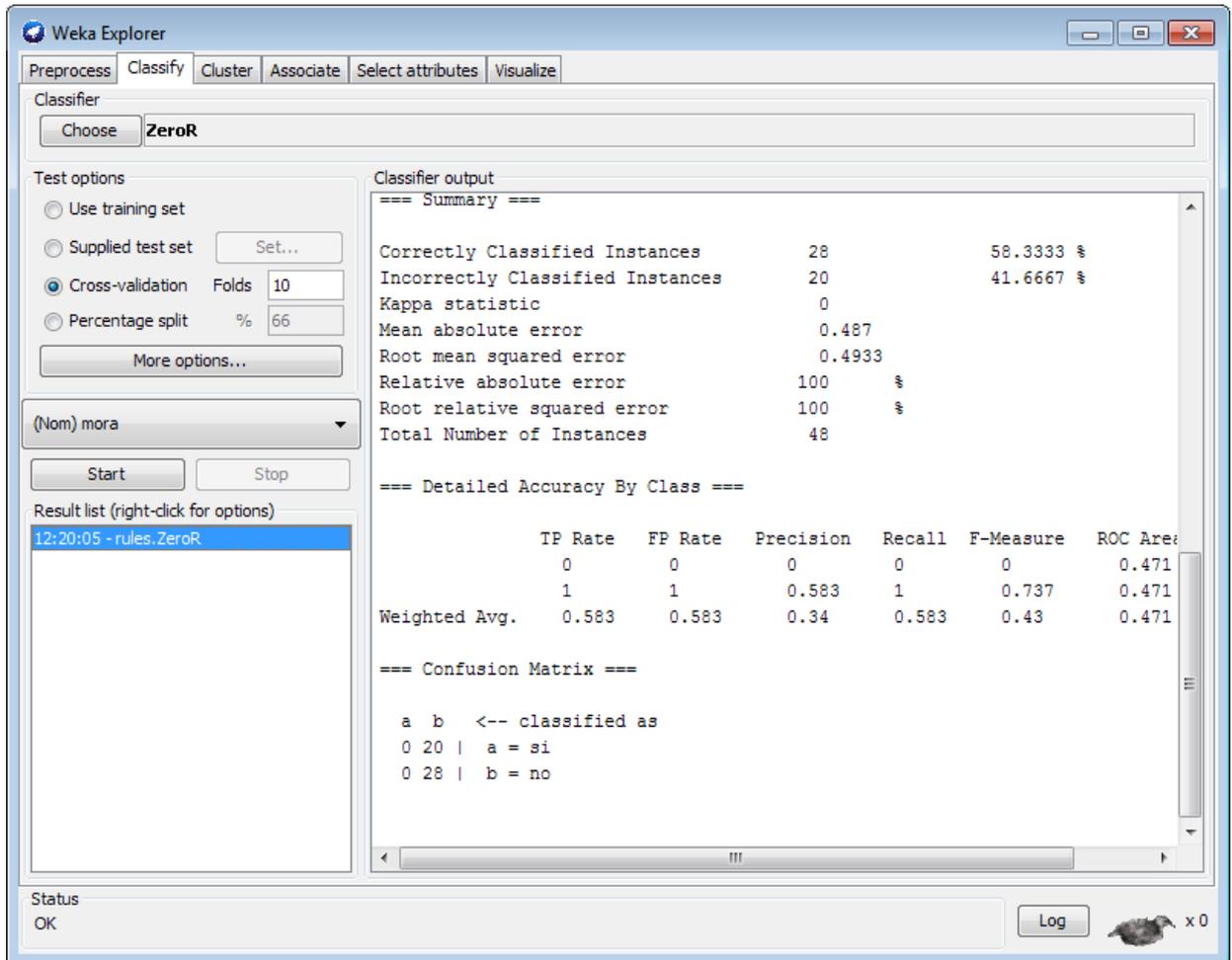


Figura 19. Resultados del clasificador.

Fuente: resultado de ejecución de Classify.

Dentro de Weka tenemos una amplia variedad de Clasificadores¹³, podemos escoger el que nos presente más aciertos en la clasificación de nuestros datos.

4.4. Algoritmos de Clasificación en Weka.

4.4.1. Algoritmo J48.

Este algoritmo es uno de los más utilizados en Weka para la minería de datos basado en arboles de decisión; se trata de un clasificador sencillo, rápido y ha dado buenos resultados en comparación con otros algoritmos más complejos; una de sus características principales es que selecciona el atributo o atributos que mejor explican la clase de salida.

¹³ Existen clasificadores que son más efectivos de acuerdo al tipo de datos, se tiene que ir verificando el que mejores resultados presente.

El algoritmo J48 de Weka es derivado del C45, el cual fué desarrollado por Ross Quinlan y es utilizado para generar un árbol de decisión para motivos de clasificación utilizando la entropía de la información, la cual se considera como una medida del desorden o la peculiaridad de ciertas combinaciones; de este modo tomando en cuenta la información promedio que contienen las datos se tiene que los datos con menor probabilidad de ocurrencia son los que aportan mayor información, por ejemplo dentro de un texto, existe mayor cantidad de artículos que nombres propios, si se borra un artículo se comprenderá la oración, no así si se borra una palabra como por ejemplo; casa, árbol, hombre.

Vamos a utiliza éste algoritmo con un folder de 10 participaciones, con lo cual podemos observar que el porcentaje de aciertos subió a 34 (70.83 %) frente a 14 (29.16 %) de errores; además la Matriz de Confusión nos muestra los datos acertados en la diagonal (14 y 20). Verificando el resultado del algoritmo J48 (Figura 20) comprobamos que nos da mejores resultados que el ZeroR.

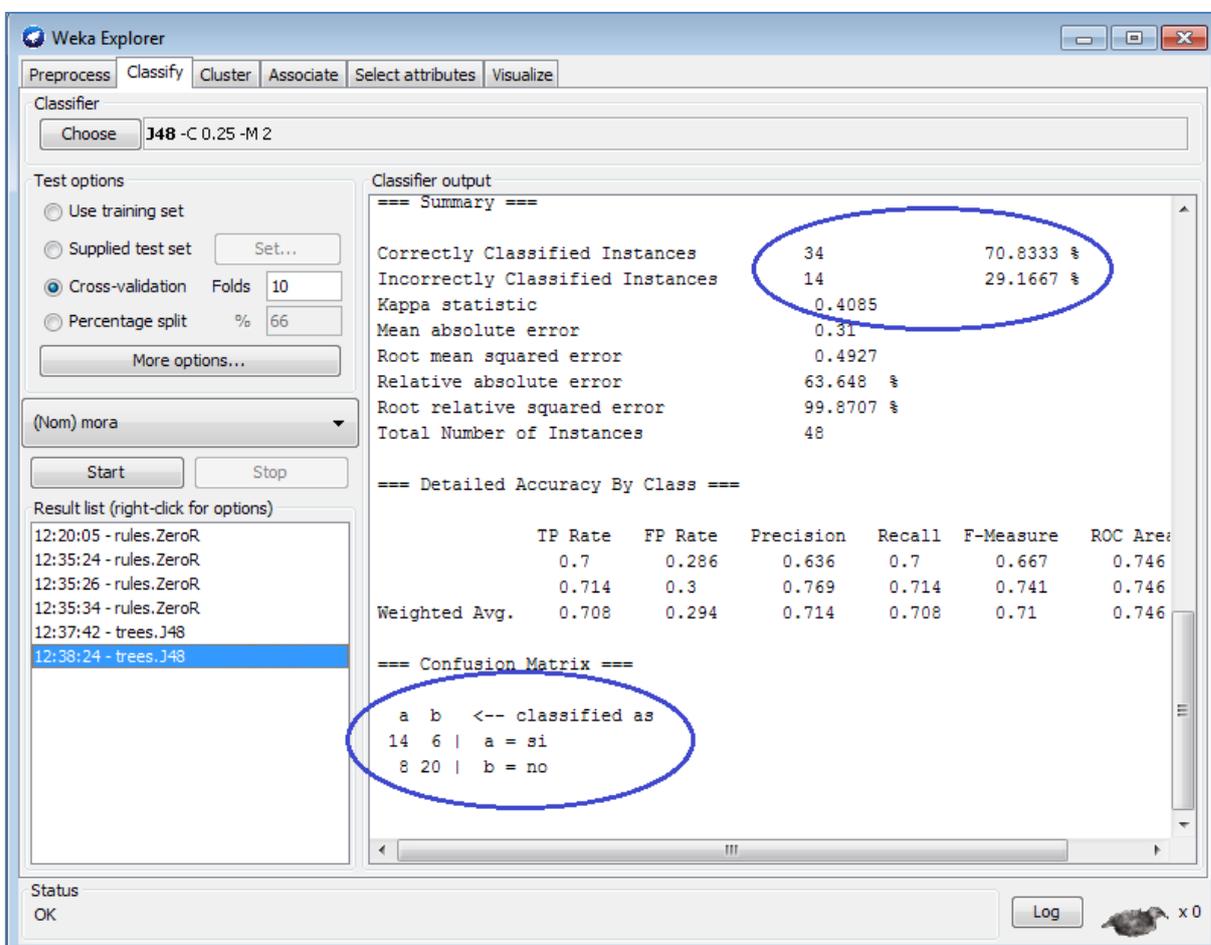


Figura 20. Resultado del algoritmo J48.

Fuente: resultado de ejecución de Classify, algoritmo J48.

En cuanto a los datos presentados en el cuadro anterior vemos que existe un porcentaje de errores altos o de instancias clasificadas incorrectamente, nuestro trabajo consiste en hacer que dichos errores disminuyan para ello vamos a aumentar el número de folder a 12 por ejemplo. Los resultados obtenidos nos indican que el porcentaje de aciertos de J48 (Figura 21) subió a un total de 36 instancias, lo que nos da un 75 % frente a 12 instancias mal clasificadas igual a 25 %.

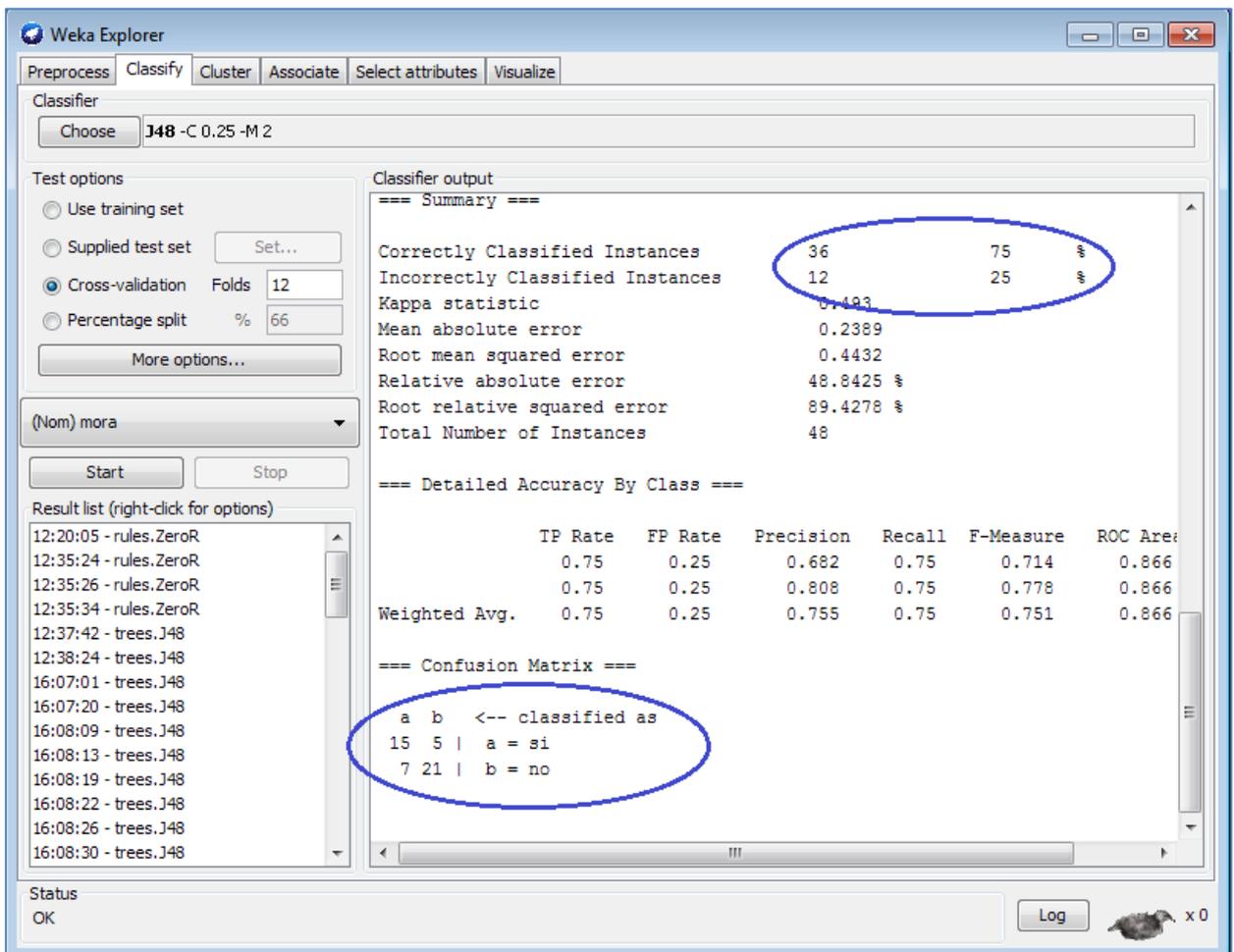


Figura 21. Porcentaje de aciertos de J48.

Fuente: resultado de ejecución de Classify, algoritmo J48.

Realizando las comprobaciones con un número de folders pertinentes (Figura 22), verificamos que el número 34 es el que mejores resultados nos presenta un total de 40 instancias con aciertos, con un 83.33 % frente a 8 instancias mal clasificadas con un valor igual a 16.67 %.

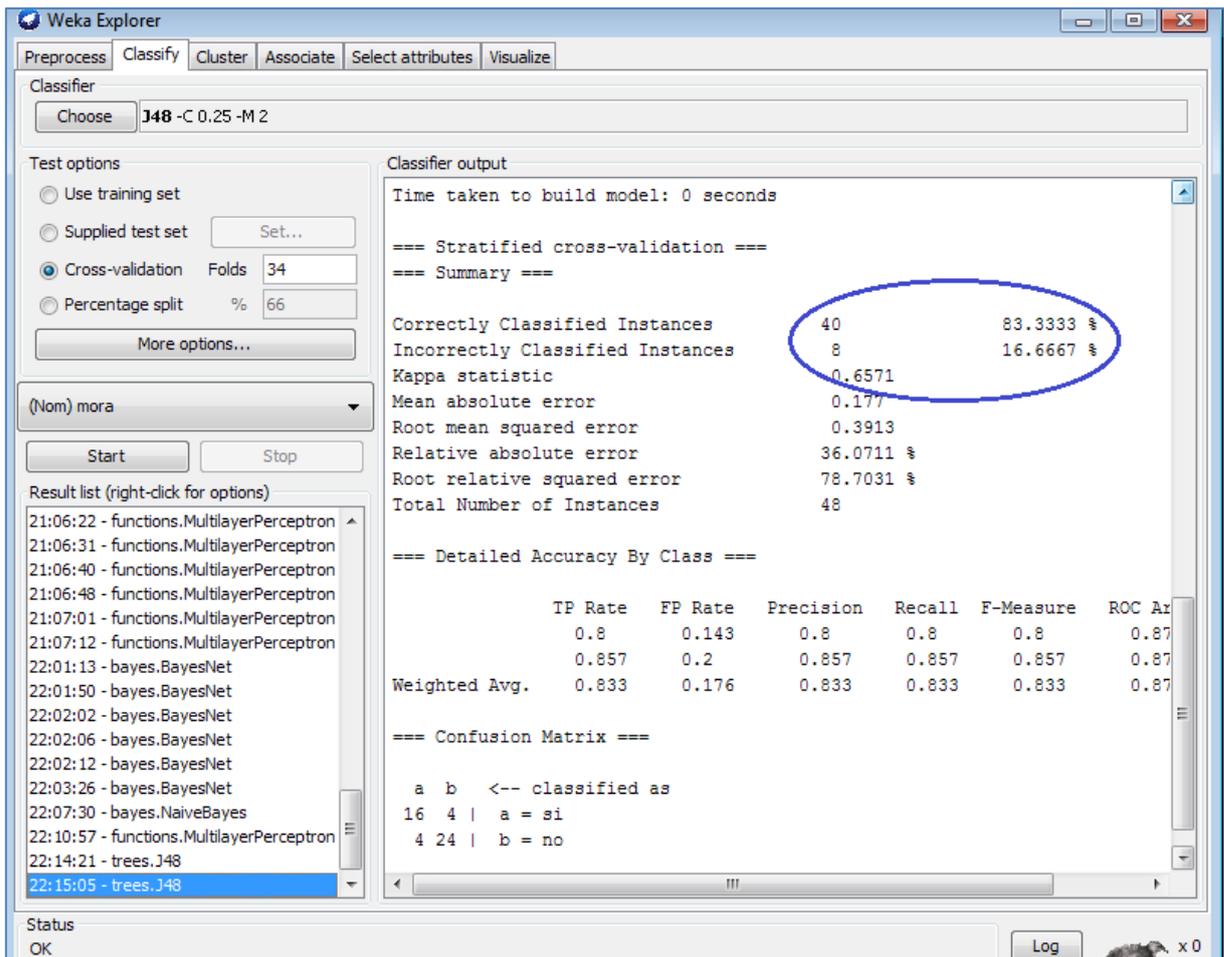


Figura 22. Comprobaciones con un número de folders.

Fuente: resultado de ejecución de Classify, algoritmo J48.

Hemos visto en los resultados anteriores, que los porcentajes de errores bajaron, pero tenemos que minimizarlos lo más que se pueda, para ellos vamos a seguir verificando otros clasificadores en la herramienta Weka.

En la parte inferior izquierda, con el botón derecho del mouse existe opciones de visualización (Figura 23), en donde podemos escoger la opción de visualización del árbol de clasificación (Figura 24) con todos sus valores, nodos y ramas, el cual también lo podemos visualizar en la sección de Classifier Output en forma de una secuencia condicional IF si nos ubicamos al inicio.

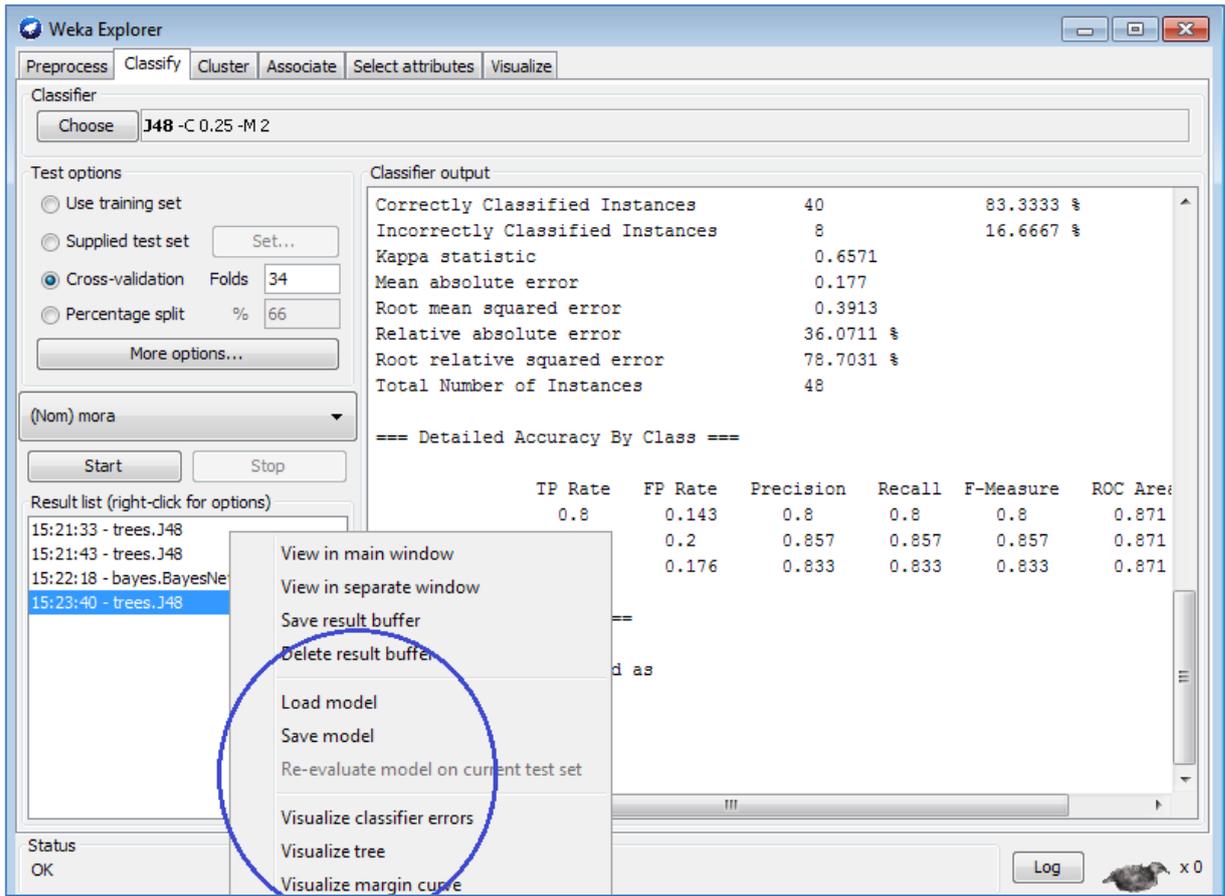


Figura 23. Opciones de visualización.

Fuente: resultado de ejecución de Classify, algoritmo J48.

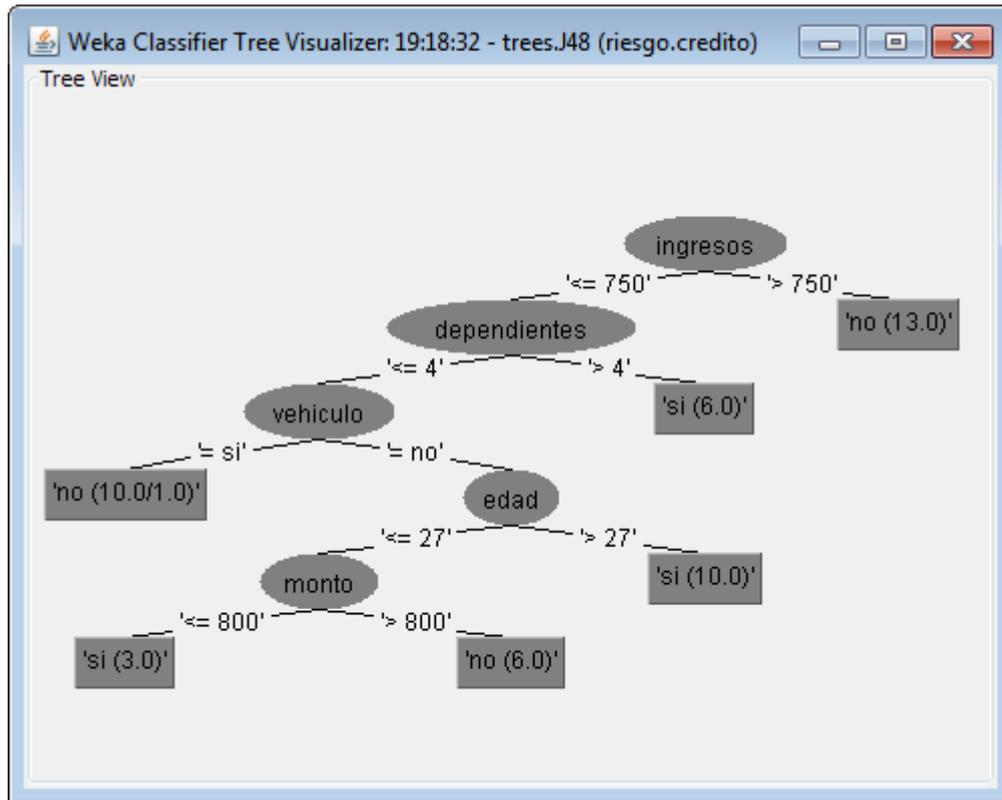


Figura 24. Visualización del árbol de clasificación.

Fuente: resultado de ejecución de Classify, algoritmo J48.

4.4.2. Perceptrón Multicapa.

El Perceptrón Multicapacon Weka (Figura 25) es un buen clasificador y se utiliza bastante en trabajos de Data Mining, pero en el análisis de nuestro trabajo nos dio un resultado un tanto inferior a J48 y aunque se aproximó bastante a éste último su velocidad de procesamiento es un poco más lenta¹⁴.

¹⁴ El Perceptrón Multicapa presenta mayor complejidad de procesamiento de información que el algoritmo J48 ya que trabaja con pesos y aplica redes neuronales a los atributos.

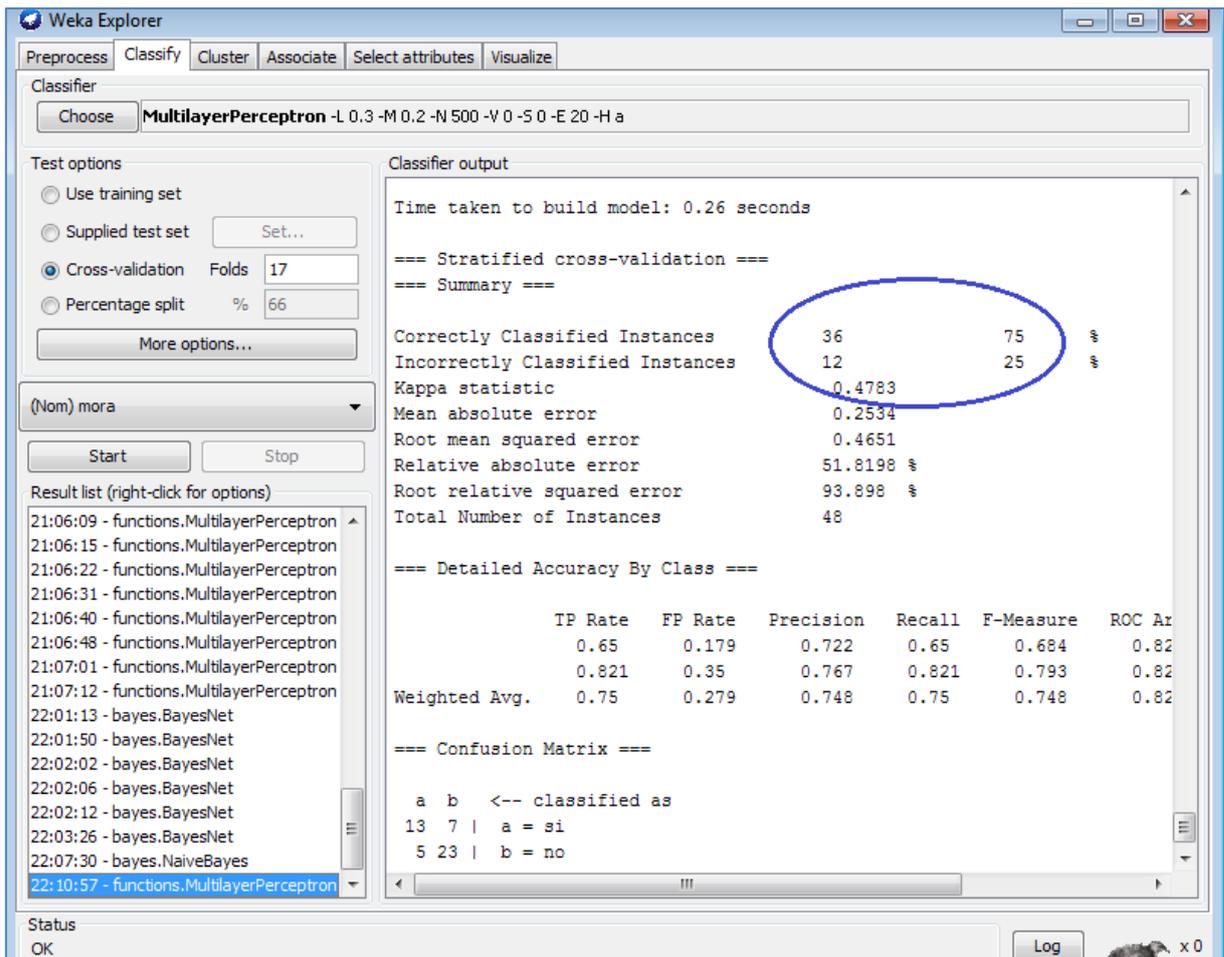


Figura 25. Perceptrón Multicapa con Weka.

Fuente: resultado de ejecución de Perceptrón Multicapa.

4.4.3. BayesNet y NaiveBayes.

El algoritmo **BayesNet**, red Bayesiana o modelo probabilístico representa un conjunto de variables aleatorias y sus dependencias condicionales a través de un grafo acíclico dirigido (DAG por sus siglas en inglés). Por ejemplo, una red bayesiana puede representar las relaciones probabilísticas entre enfermedades y síntomas. Dados los síntomas, la red puede ser usada para computar la probabilidad de la presencia de varias enfermedades.

Aplicando la red bayesiana en Weka, así mismo como en los algoritmos anteriores escogemos el número de folders que mejor nos dé resultados (Figura 26), para lo cual el número 15 nos presenta un total de 36 instancias con aciertos, con un 75 % frente a 12 instancias mal clasificadas con un valor igual a 25 %.

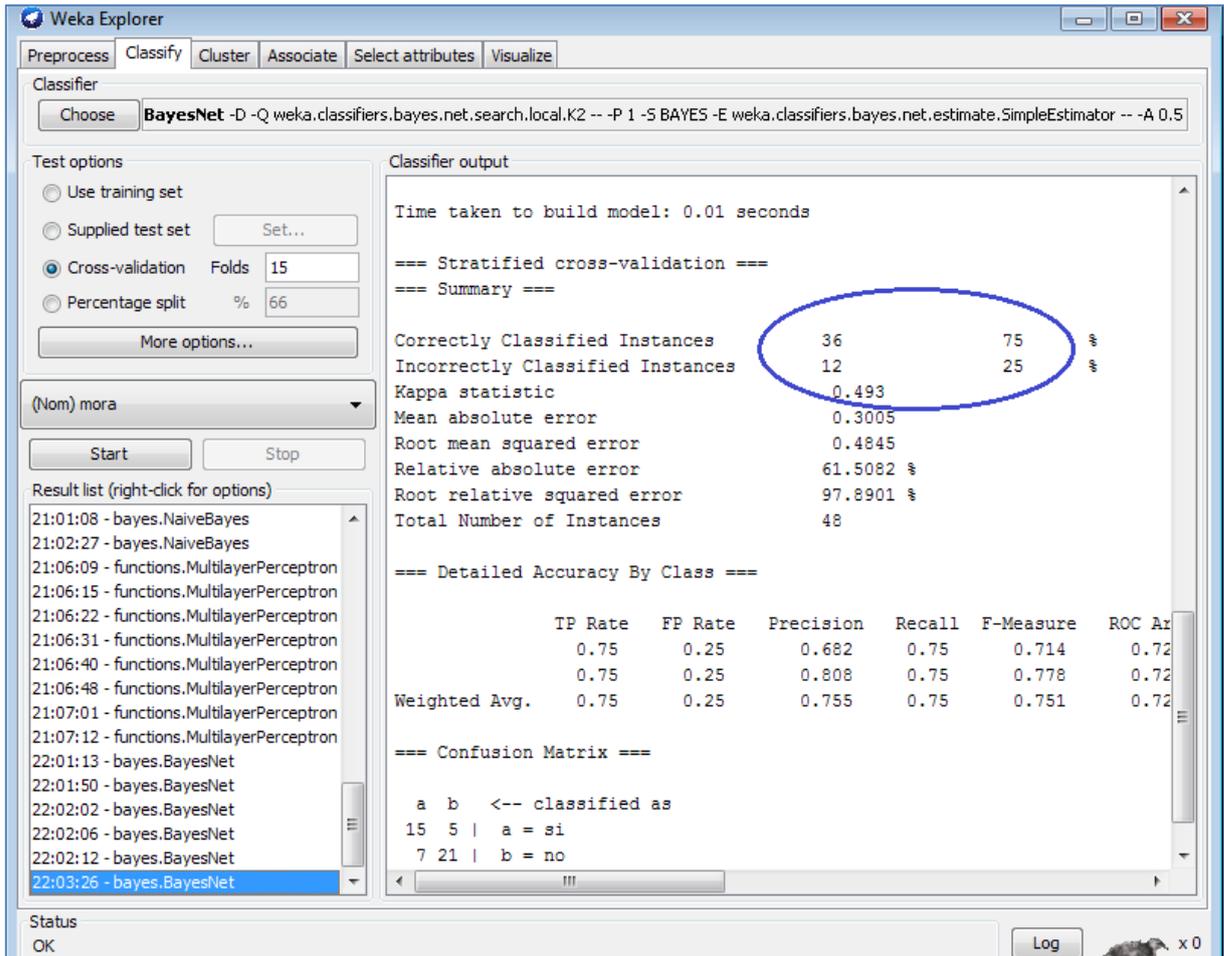


Figura 26. BayesNet con Weka.

Fuente: resultado de ejecución de BayesNet.

Seguidamente vamos a revisar el algoritmo **NaiveBayes** conocido como uno de los clasificadores más utilizados por su simplicidad y rapidez, tratándose de una técnica de clasificación y predicción supervisada que construye modelos que predicen la probabilidad de posibles resultados.

Aplicando el algoritmo (Figura 27), tenemos que para 14 folders nos presenta un total de 35 instancias con aciertos, con un 72.9167 % frente a 13 instancias mal clasificadas con un valor igual a 27.0833 %.

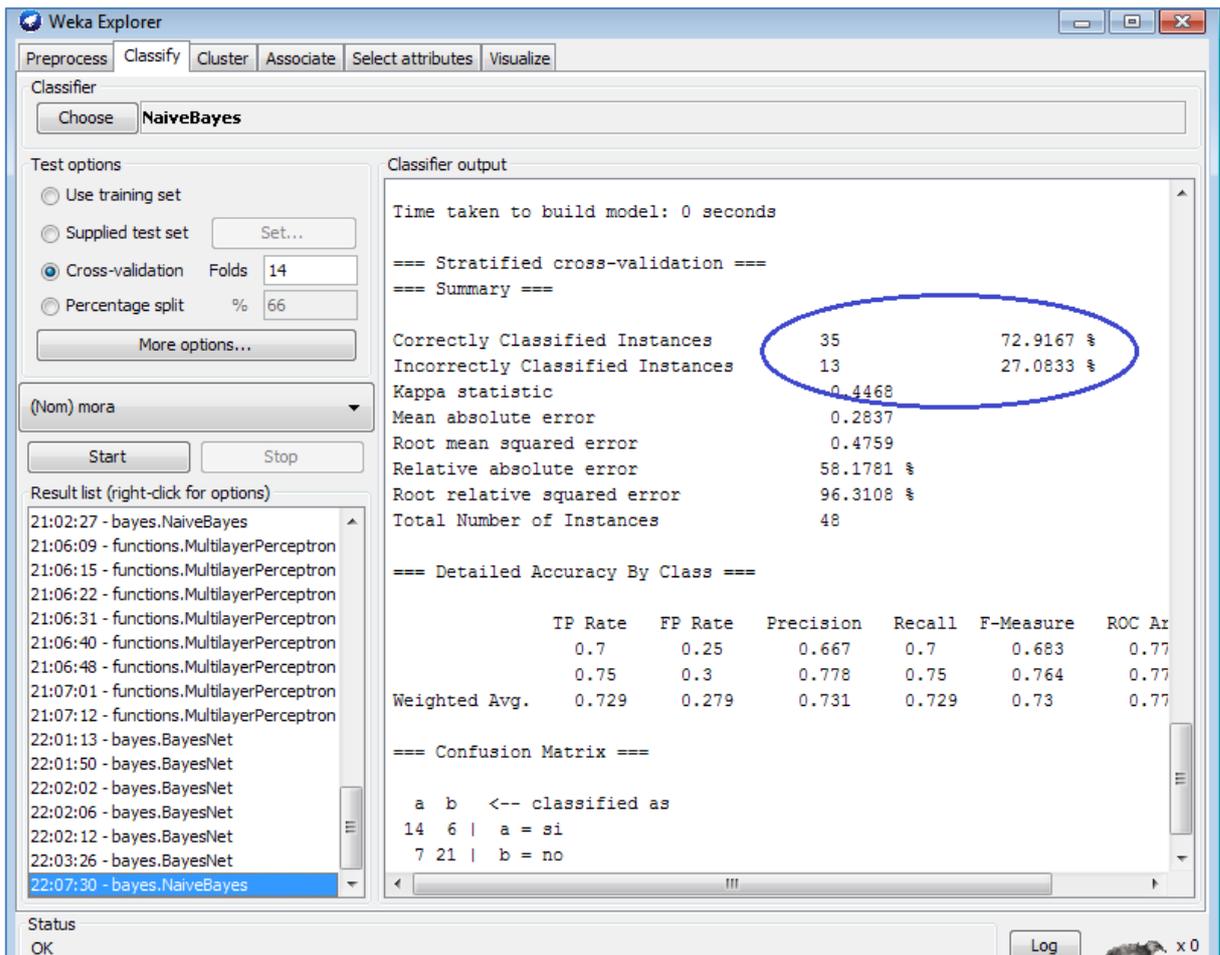


Figura 27. NaiveBayes con Weka.

Fuente: resultado de ejecución de NaiveBayes.

4.4.4. Comparación de resultados de los algoritmos.

Para poder verificar cual es el algoritmo que mejor resultados nos ha presentado de los anteriormente revisados, vamos a tomar en consideración las siguientes parámetros de Weka: Número de folds más adecuado, Tiempo Tomado (Time taken to build model), Correctly Classified Instances (CCI), Incorrectly Classified Instances (ICI), Mean absolute error (error absoluto promedio MAE) y Root Mean squared error (error cuadrado promedio RMSE); en la siguiente tabla vamos a presentar la comparación de los resultados de los algoritmos (Tabla 3):

Tabla 3. Comparación de resultados de los algoritmos.

PARÁMETROS	ALGORITMOS DE CLASIFICACIÓN			
	J48	PERCEPTRON ML	BayesNet	NaiveBayes
folds	34	17	15	14
Time Taken	0	0.26	0.01	0
CCI	40	36	36	35
ICI	8	12	12	13
MAE	0.177	0.2534	0.3005	0.2837
RMSE	0.3913	0.4651	0.4845	0.4759

Fuente: construcción propia

Según la verificación de los clasificadores, se ha determinado que el **J48 presentó mejores resultados**, por las siguientes razones:

- Menor tiempo tomado en construir el modelo.
- Mayor número de instancias correctamente clasificadas.
- Menor porcentaje de error en MAE y RMSE.

Si hacemos una comparación tomando un número de folds promedios para el cálculo (Tabla 4), por ejemplo 17, aun así podemos verificar que el algoritmo J48 sigue presentando mejores resultados que el resto:

Tabla 4. Comparación de resultados con 17 folds.

PARÁMETROS	ALGORITMOS DE CLASIFICACIÓN			
	J48	PERCEPTRON ML	BayesNet	NaiveBayes
folds	17	17	17	17
Time Taken	0	0.26	0.01	0.01
CCI	38	36	35	35
ICI	10	12	13	13
MAE	0.226	0.2534	0.2958	0.2921
RMSE	0.4192	0.4651	0.4779	0.4883

Fuente: construcción propia

4.5. Selección de atributos relevantes.

Mediante la selección de atributos en Weka (Figura 28) vamos a eliminar los que nos resulten redundantes e irrelevantes; ya que al existir un número elevado de atributos puede ser que dentro del modelo se produzca complejidad y no nos conduzca a la obtención del resultado esperado. Escogiendo los atributos adecuados podemos hacer que el modelo sea más efectivo y sencillo. Dentro de Weka podemos escoger las tres posibilidades:

- a. Evaluación de atributos: con el Método de búsqueda = Ranker y Método de Evaluación = InfoGainAttributeEval.

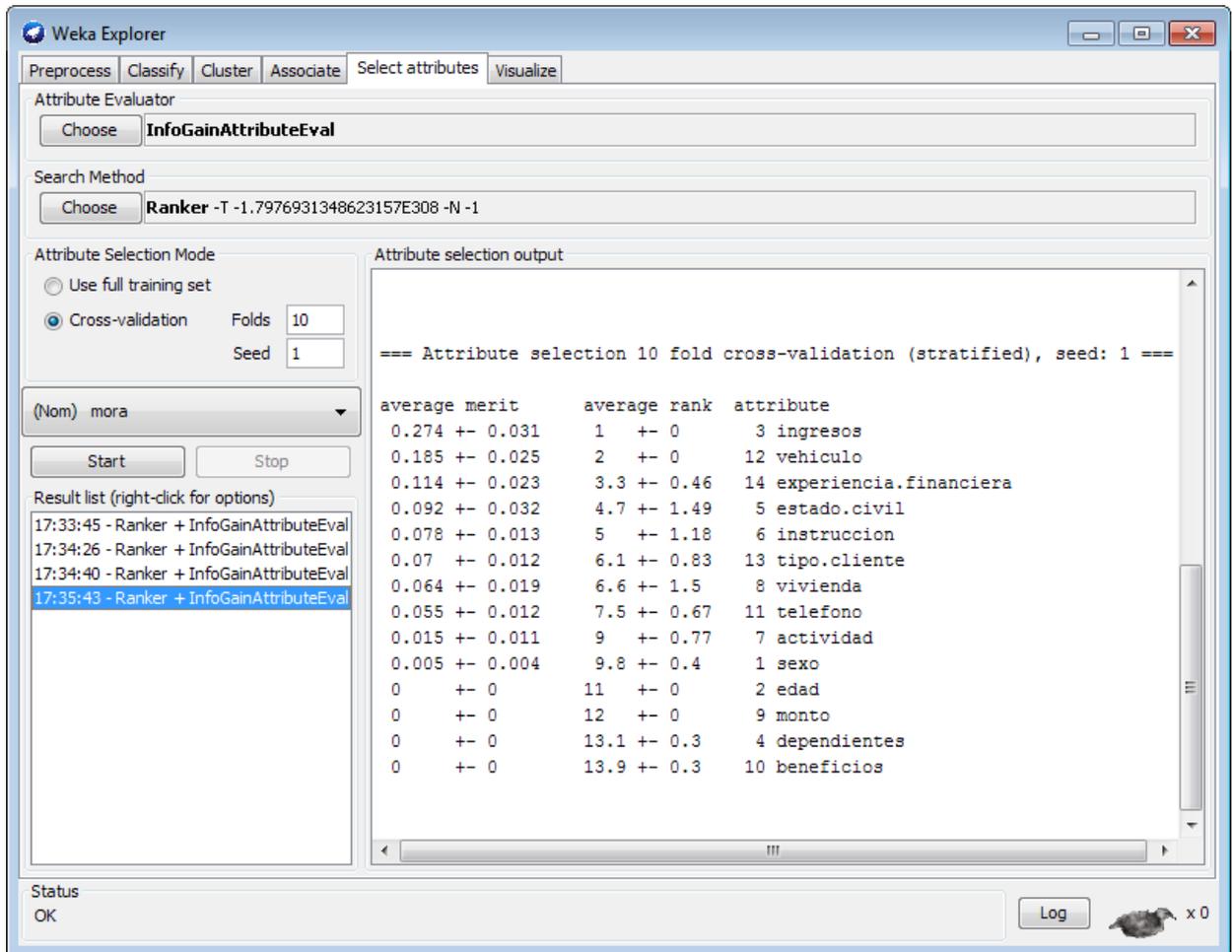


Figura 28. Selección de atributos en Weka.

Fuente: resultado de ejecución de Select Attributes.

Podemos ver que en la tabla anterior que se ha utilizado un cross-validation de 10 hojas para obtener mayor precisión y en la columna de average podemos verificar los atributos con mayor puntaje (Tabla 3), por ejemplo:

Tabla 5. Atributos con mayor puntaje.

average merit	average rank	attribute
0.274 +- 0.031	1 +- 0	3 ingresos
0.185 +- 0.025	2 +- 0	12 vehículo
0.114 +- 0.023	3.3 +- 0.46	14 experiencia.financiera
0.092 +- 0.032	4.7 +- 1.49	5 estado civil
0.078 +- 0.013	5 +- 1.18	6 instrucción
0.07 +- 0.012	6.1 +- 0.83	13 tipo.cliente

Fuente: construcción propia

Según estos resultados vamos a seleccionar los 6 atributos para obtener un mejor resultado en nuestro modelo. En un conjunto de datos con muchos atributos es aconsejable utilizar *use full training set*.

Con el fin de comprobar resultados vamos a realizar la otra prueba de selección de atributos:

- b. Metodo Filter: con el Método de búsqueda = Greedy Stepwise y Método de evaluación = CfsSubsetEval. (Figura 29).

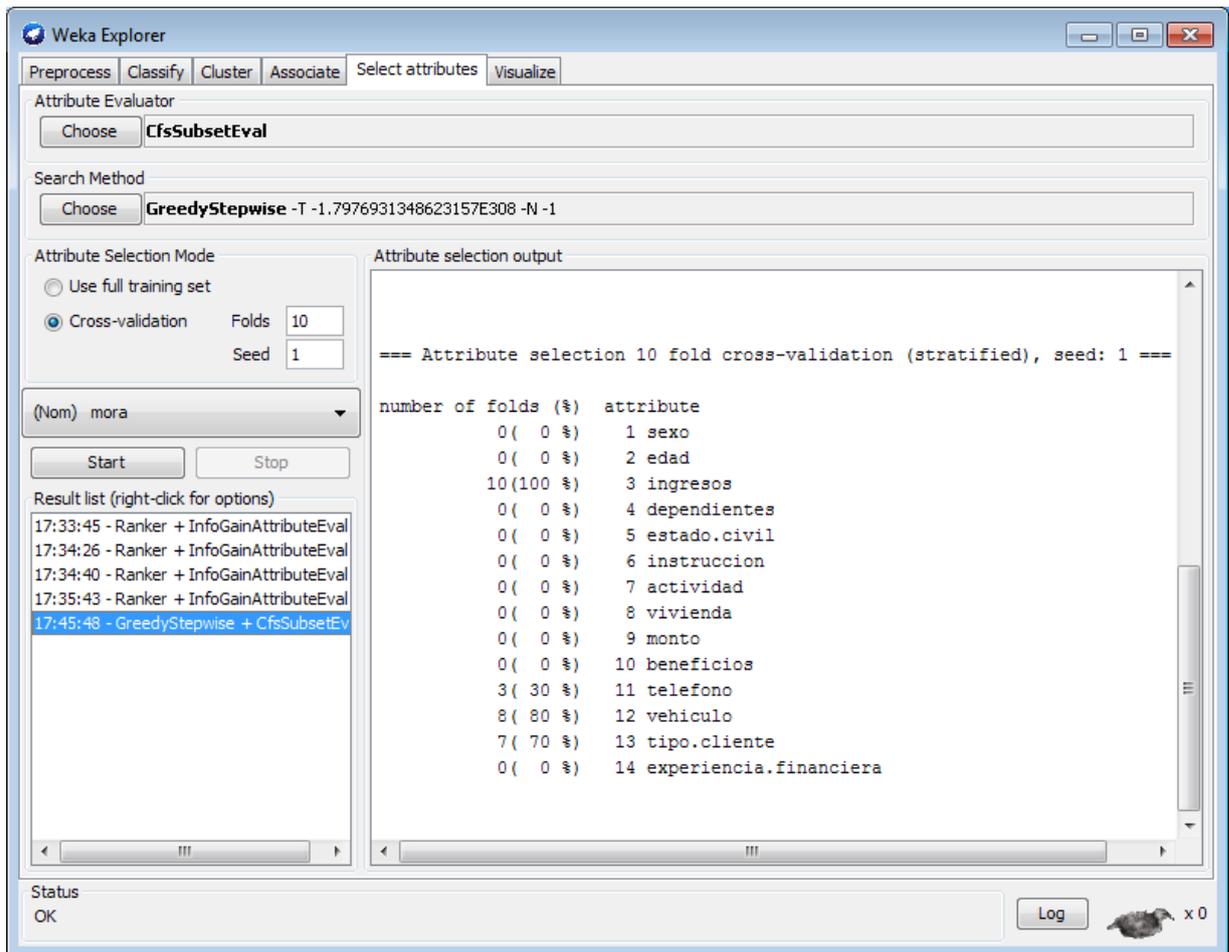


Figura 29. Método Filter.

Fuente: resultado de ejecución de Select Attributes.

Ahora podemos notar que los atributos con mayor puntaje son:

Ingresos = 100 %, tipo cliente = 70 %, vehículo = 80 %, teléfono = 30 %

Por último vamos a seleccionar otro método de selección de atributos:

- c. Método Wrapper: Método de búsqueda = Greedy Stepwise y Método de evaluación = ClassifierSubsetEval (Figura 30).

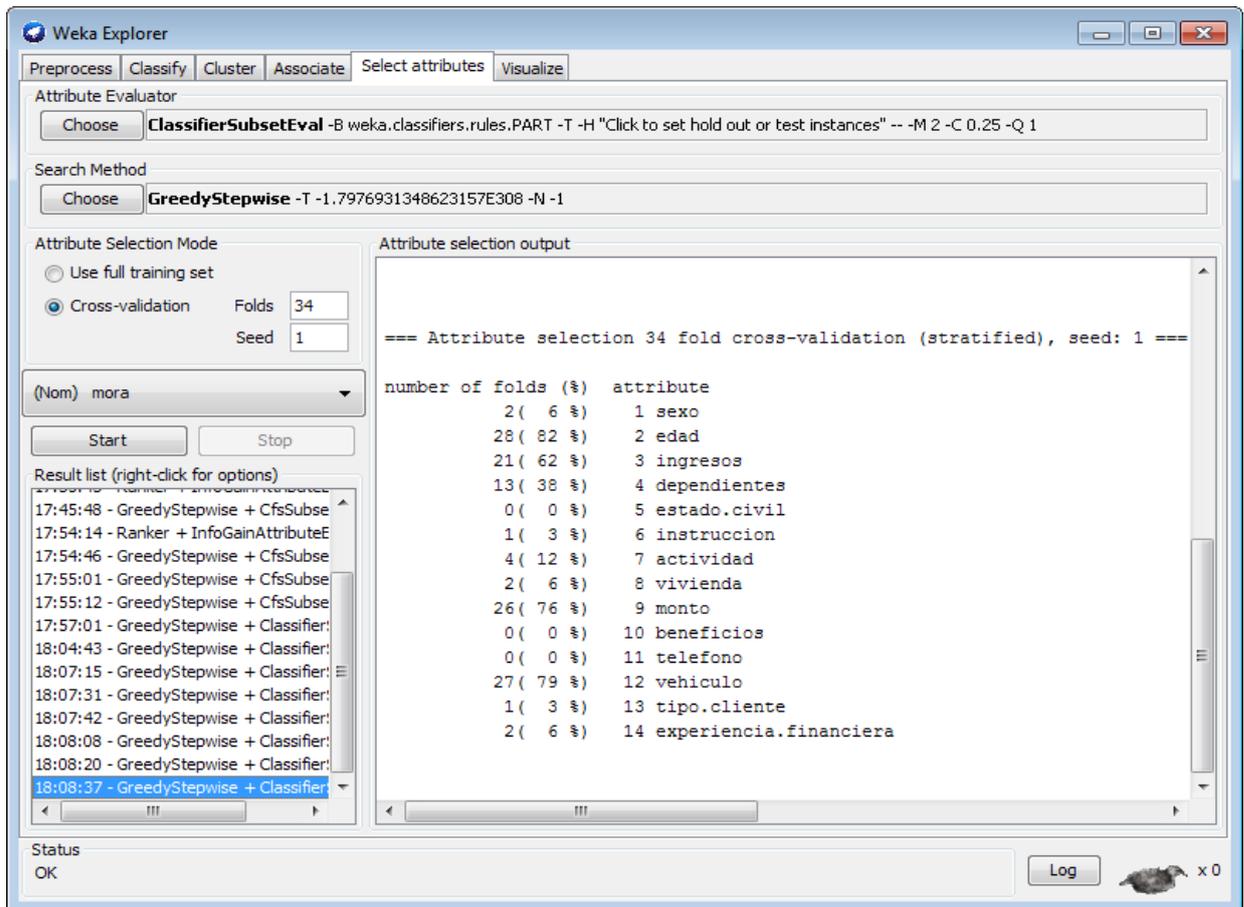


Figura 30. Método Wrapper.

Fuente: resultado de ejecución de Select Attributes.

Dentro de ClassifierSubsetEval tenemos que elegir un clasificador base, para ello hemos utilizado el PART; lo cual nos indica que los siguientes atributos tiene el mejor puntaje:

edad = 82 %, vehículo = 79 %, monto = 76 %, ingresos = 62 %, dependientes = 38 %,

En base a las tres pruebas anteriores de selección, vamos a escoger los atributos de mejor puntaje (Figura 31) y finalmente vamos a trabajar con éstos para nuestro presente trabajo:

- Ingresos
- Vehículo
- Edad
- Dependientes

- Tipo de Cliente
- Monto
- Mora (ultimo atributo)

Seguidamente, dentro de la herramienta vamos a borrar los atributos redundantes y solamente vamos a trabajar con los escogidos para realizar la comprobación:

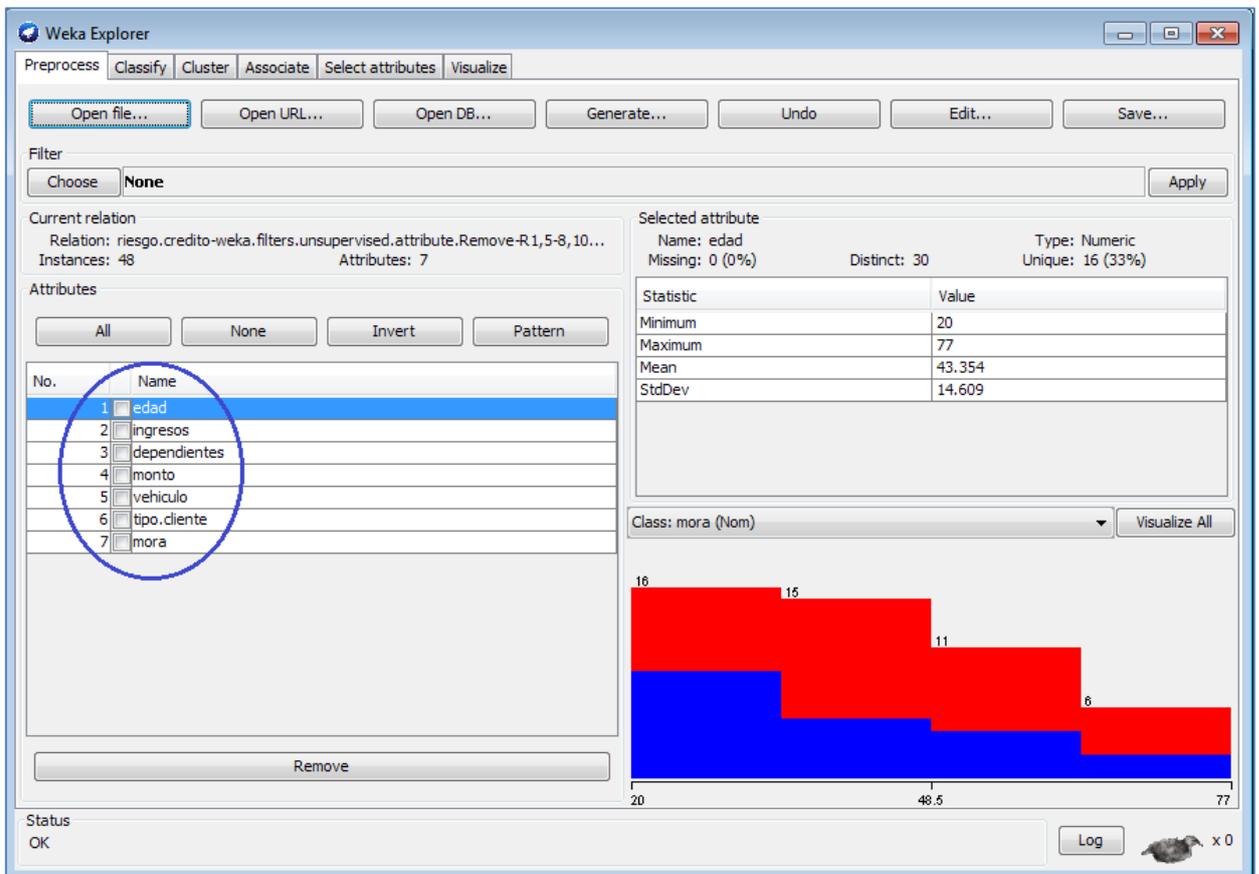


Figura 31. Atributos de mejor puntaje.

Fuente: resultado de verificación de Preprocess.

Al realizar la clasificación se obtuvo el siguiente resultado con el algoritmo J48 (Figura 32):

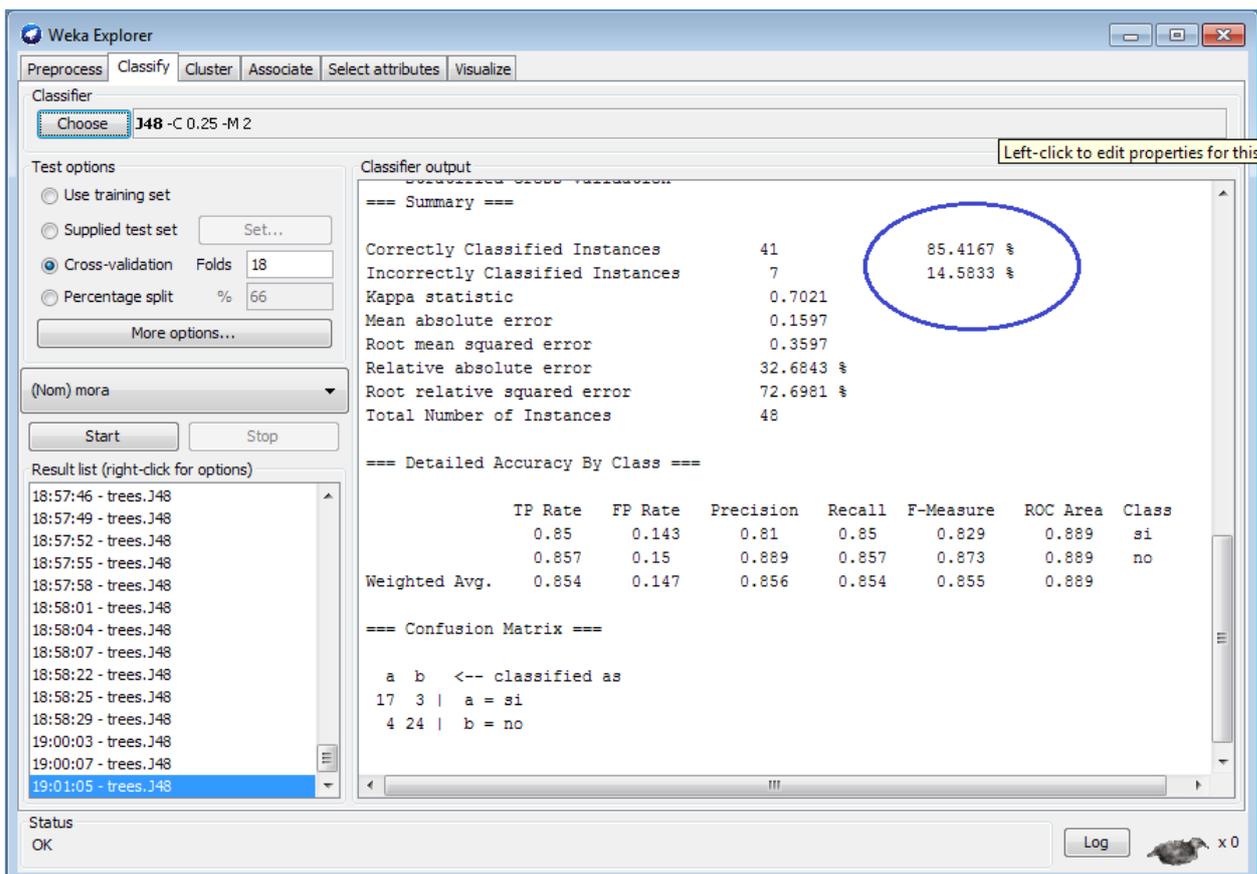


Figura 32. Resultado con el algoritmo J48.

Fuente: resultado de ejecución de Classify.

El valor obtenido ahora es 85.4167 % de acierto, en relación al 83.333 % anterior cuando aún no se realizaba la eliminación de los atributos; por lo tanto hemos obtenido un mejor resultado, lo cual era lo que se esperaba.

En el cuadro anterior podemos verificar que los atributos no clasificados correctamente en la matriz de confusión ahora son de 3 en la columna de no y 4 en la columna de si, lo cual mejor su porcentaje; además si en la ventana subimos hacia la parte inicial podemos verificar el árbol de decisión (Figura 33) y las reglas tomadas en cuenta:

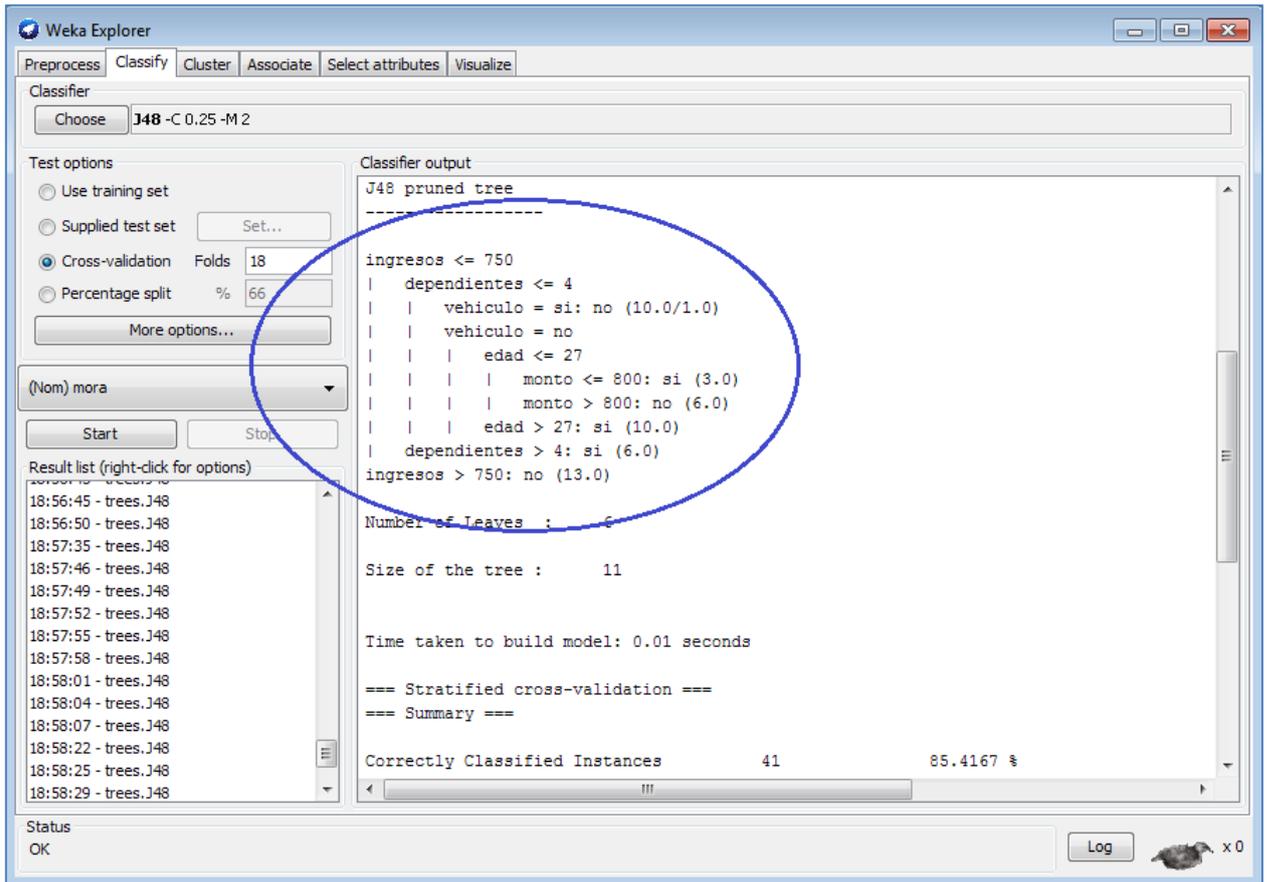


Figura 33. Árbol de decisión.

Fuente: resultado de ejecución de Classify.

Un aspecto importante dentro de la herramienta de Weka es la visualización de resultados (Figura 34); para ello nos vamos a la pestaña de Visualize, en donde vamos a poder determinar la relación de los atributos por parejas:

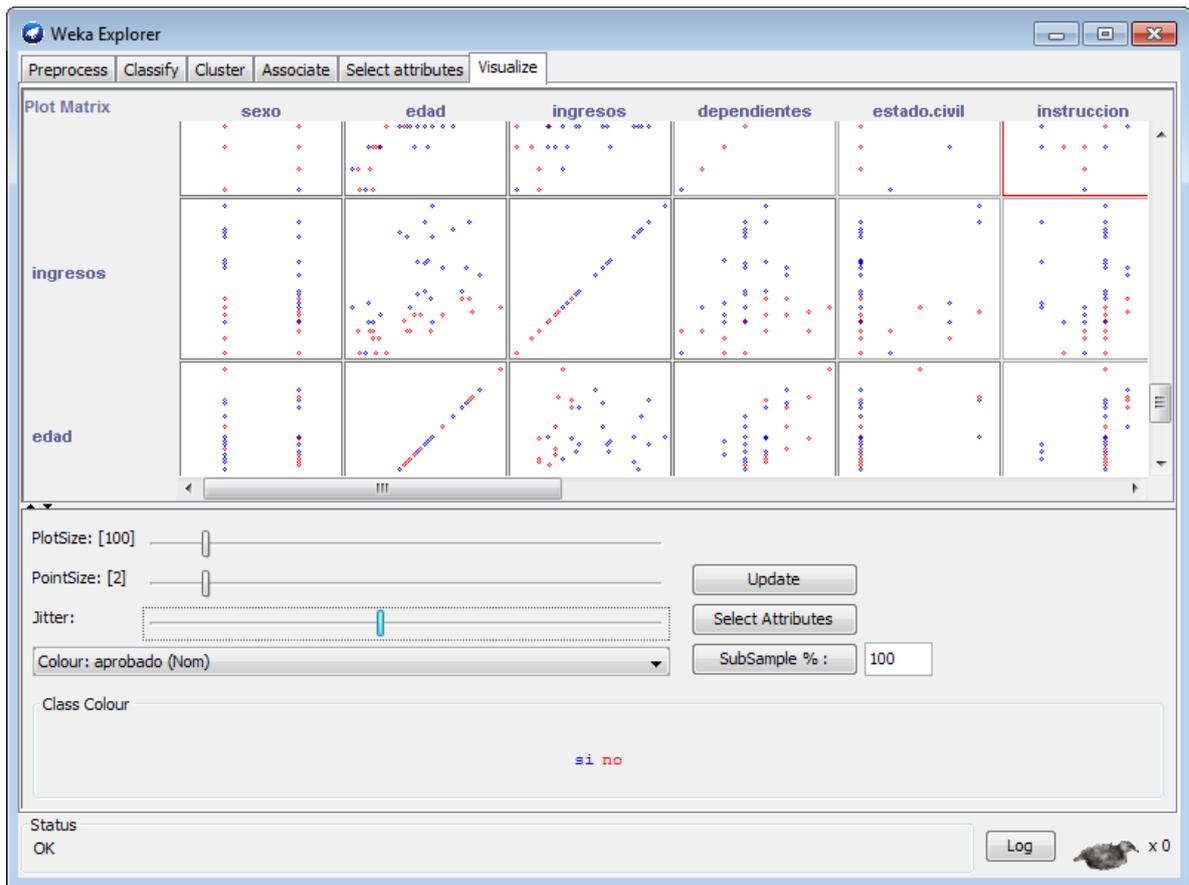


Figura 34. Visualización de resultados.

Fuente: resultado de ejecución de Visualize.

En esta ventana vamos a verificar un par atributos relevantes (Figura 35), por ejemplo: *ingresos* y *dependientes*, en donde podemos ver que se encuentran casi perfectamente separados por lo que se deduce que son factibles de clasificación:

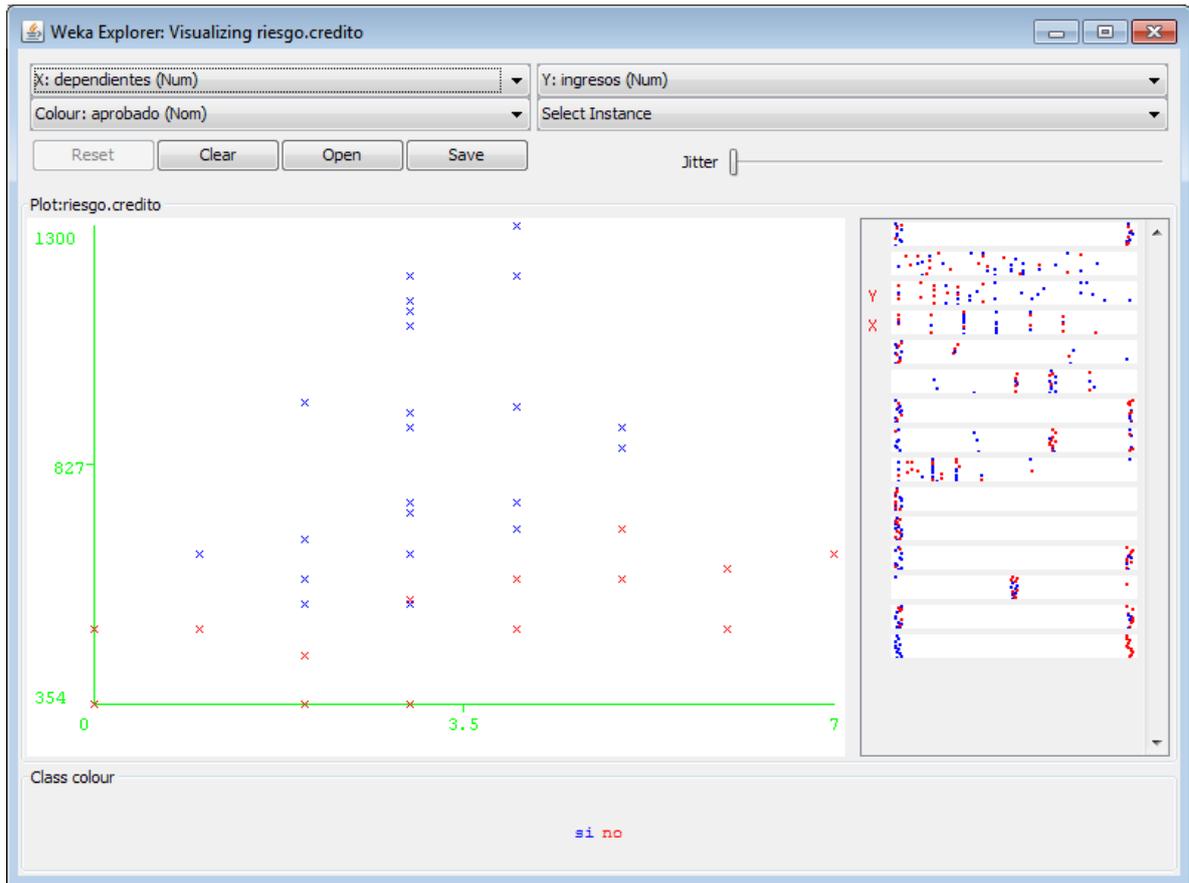


Figura 35. Par de atributos relevantes.

Fuente: resultado de ejecución de Visualize.

4.6. Validación de resultados del modelo predictivo.

Una vez que nosotros hemos obtenido los resultados aplicando el modelo con la ayuda de la herramienta weka, vamos a realizar un análisis con los resultados reales obtenidos dentro de la cooperativa, tomando en cuenta que clientes se encuentran en mora y cuáles no, para de ésta manera poder evaluar el funcionamiento de nuestro modelo y poder hacer un ajuste necesario de acuerdo a la realidad.

De acuerdo al resultado del modelo presentado anteriormente:

```

ingresos <= 750
  dependientes <= 4
    vehiculo = si: no (10)
    vehiculo = no
      edad <= 27
        monto <= 800: si (3)
        monto > 800: no (6)
      edad > 27: si (10)
    dependientes > 4: si (6)
  ingresos > 750: no (13)

```

Significa que, de 48 casos analizados:

- Si el cliente tiene un ingreso menor o igual a \$ 750: entonces:
 - Si el número de dependientes es menor o igual a 4: entonces:
 - Si posee vehículo: (no en mora 10 casos)
 - Si no posee vehículo: entonces:
 - Si edad es menor o igual a 27: entonces:
 - Si monto es menor o igual a 800: (si en mora 3 casos)
 - Si monto es mayor a 800: (no en mora 6 casos)
 - Si edad es mayor a 27: (si en mora 10 casos)
 - Si el número de dependientes es mayor a 4: (si en mora 6 casos)
 - Si el cliente tiene un ingreso mayor a \$ 750: (no en mora 13 casos)

Tomando en cuenta lo anterior se verificaron todos los atributos que se encuentran en mora de acuerdo a los valores (Tabla 4) proporcionados por la institución financiera.

Tabla 6. Atributos con valores en mora.

ATRIBUTOS	MORA
M,41,1100,3,casado,secundaria,primaria,hipotecada,3000,IESS,celular,si,socio,si,si	
F,48,750,3,casado,secundaria,secundaria,propia,2000,IESS,celular,si,normal,si,si	
F,44,600,2,casado,secundaria,secundaria,propia,1500,IESS,celular,si,normal,si,si	
F,27,500,1,casado,estudiante,secundaria,arrendada,1000,IESS,celular,no,normal,no,no	
F,48,1200,3,casado,superior,secundaria,hipotecada,1500,IESS,celular,si,normal,si,si	
M,59,1150,3,casado,secundaria,primaria,propia,1500,IESS,celular,si,normal,si,si	

F,25,354,0,soltero,estudiante,secundaria,arrendada,600,IESS,celular,no,normal,no,no	361,76
M,42,650,3,casado,superior,primaria,arrendada,1000,IESS,celular,si,normal,si,si	
M,44,600,2,casado,secundaria,primaria,compartida,300,IESS,celular,si,normal,si,si	
F,50,700,4,casado,secundaria,secundaria,propia,300,IESS,celular,si,normal,si,si	
M,55,900,3,casado,primaria,secundaria,propia,1500,IESS,celular,no,normal,si,si	
F,51,550,3,casado,secundaria,primaria,propia,1500,IESS,celular,si,normal,si,si	
F,42,500,4,casado,secundaria,secundaria,arrendada,500,IESS,celular,no,normal,si,no	500,00
F,65,1200,4,divorciado,secundaria,secundaria,propia,1000,IESS,celular,si,normal,si,si	
F,28,500,0,soltero,estudiante,primaria,arrendada,1545,IESS,celular,no,externo,no,no	957,22
M,30,450,2,casado,secundaria,secundaria,arrendada,1500,IESS,celular,no,normal,no,no	1.500,00
M,31,600,2,casado,estudiante,secundaria,compartida,300,IESS,celular,no,normal,no,si	201,79
F,50,500,6,casado,secundaria,secundaria,hipotecada,2000,IESS,celular,si,normal,si,no	1.774,78
M,28,550,2,casado,estudiante,secundaria,compartida,300,IESS,celular,si,normal,no,si	
F,23,354,0,soltero,estudiante,secundaria,compartida,478,IESS,celular,no,normal,no,no	199,20
M,26,550,2,union_libre,tecnico,secundaria,arrendada,1100,IESS,celular,no,normal,no,si	
M,23,354,0,soltero,estudiante,secundaria,arrendada,700,IESS,celular,no,normal,no,no	367,55
F,44,600,4,casado,secundaria,secundaria,hipotecada,4548,IESS,normal,no,externo,si,no	2.121,41
M,49,950,2,casado,secundaria,primaria,hipotecada,1000,IESS,celular,si,normal,si,si	
F,45,730,3,casado,secundaria,primaria,arrendada,2019,IESS,celular,si,normal,si,si	
M,46,600,5,casado,secundaria,secundaria,hipotecada,1000,IESS,celular,no,normal,si,no	922,71
F,51,1100,3,casado,secundaria,secundaria,propia,300,IESS,celular,si,normal,si,si	
F,41,500,4,casado,secundaria,secundaria,arrendada,500,IESS,celular,no,normal,no,no	229,74
F,69,860,5,casado,primaria,primaria,propia,1000,IESS,celular,si,normal,si,si	
M,55,600,5,casado,secundaria,primaria,propia,3015,IESS,celular,si,normal,si,no	146,19
M,77,650,7,viudo,secundaria,primaria,propia,1500,IESS,normal,no,normal,no,no	409,04
M,65,700,4,casado,primaria,secundaria,propia,1000,IESS,celular,si,normal,si,no	
M,38,1130,3,casado,secundaria,secundaria,arrendada,5000,IESS,celular,si,normal,si,si	
M,48,930,3,casado,secundaria,secundaria,propia,1000,IESS,celular,si,normal,si,si	
F,20,650,1,casado,estudiante,secundaria,arrendada,1000,IESS,celular,no,normal,no,si	
F,62,700,5,casado,primaria,primaria,hipotecada,1200,IESS,celular,si,normal,si,no	880,64
M,28,450,2,union_libre,estudiante,primaria,arrendada,1200,IESS,celular,no,normal,no,no	895,20
F,22,500,1,casado,estudiante,primaria,arrendada,1500,IESS,celular,no,normal,si,no	
F,62,750,4,casado,secundaria,secundaria,propia,1000,IESS,celular,si,normal,si,si	244,73
M,64,900,5,casado,secundaria,primaria,propia,300,IESS,celular,si,normal,si,si	
F,40,560,3,casado,secundaria,secundaria,arrendada,600,IESS,celular,no,normal,no,no	189,12
F,45,940,4,casado,superior,primaria,arrendada,300,IESS,celular,si,normal,si,si	
F,25,354,0,soltero,estudiante,secundaria,arrendada,1500,IESS,celular,no,normal,no,si	
F,26,680,2,union_libre,superior,primaria,arrendada,1000,IESS,celular,no,normal,no,si	
F,66,620,6,divorciado,primaria,primaria,propia,300,IESS,celular,si,normal,si,no	300,00
F,29,354,2,casado,tecnico,secundaria,arrendada,300,IESS,celular,no,normal,no,no	100,01
M,33,354,3,casado,secundaria,secundaria,arrendada,300,IESS,celular,no,normal,no,no	198,51
M,51,1300,4,divorciado,secundaria,secundaria,propia,800,IESS,celular,si,normal,si,si	

Fuente: construcción propia

CAPITULO V
IMPLEMENTACION DEL MODELO PREDICTIVO.

5.1. Presentación de la información.

En el capítulo anterior se realizó el análisis de los datos utilizando la herramienta Weka, con la cual se obtuvieron resultados que nos sirven para la predicción de los riesgos financieros de nuestra empresa en cuestión. Al ser Weka una herramienta de análisis sirve de gran ayuda al desarrollador para obtener la información, pero no es muy amigable al usuario final ya que debe tener un conocimiento bastante aceptable del manejo de la misma y el proceso no es automático para obtener el resultado ya que el manejo de la plataforma conlleva varios pasos como pudimos ver anteriormente.

Como hemos visto anteriormente, en Weka se presentan los resultados dentro de la herramienta, pero el usuario final que va a utilizar nuestro modelo va a querer almacenar y manipular dicha información; para resolver el tema anterior vamos a crear un ambiente amigable mediante otra herramienta informática para el usuario final utilizando los mismos procesos y obteniendo los resultados que se hacen en Weka.

5.2. Plataforma de desarrollo a utilizar: Java Netbeans, creación del proyecto.

Para poder presentar y manipular los resultados de Weka de una forma más rápida y fácil para el usuario, vamos a crear una aplicación con el lenguaje de Programación Java Netbeans, (Figura 34). Nuestro proyecto se llamará *wekadatamining* y en el Anexo 6 detallamos como construir el proyecto. Las ventajas al utilizar Java Netbeans para nuestro proyecto son las siguientes:

- Compatibilidad con la plataforma Weka
- Netbeans es libre, de código abierto, sin restricciones de uso y se la puede adquirir de forma gratuita en el internet.
- Multiplataforma¹⁵ con soporte integrado para el lenguaje de programación Java.
- Herramienta potente de gestión de interfaz de usuario

¹⁵ Programas informáticos o conceptos de cómputo que pueden funcionar en múltiples plataformas o sistemas operativos.

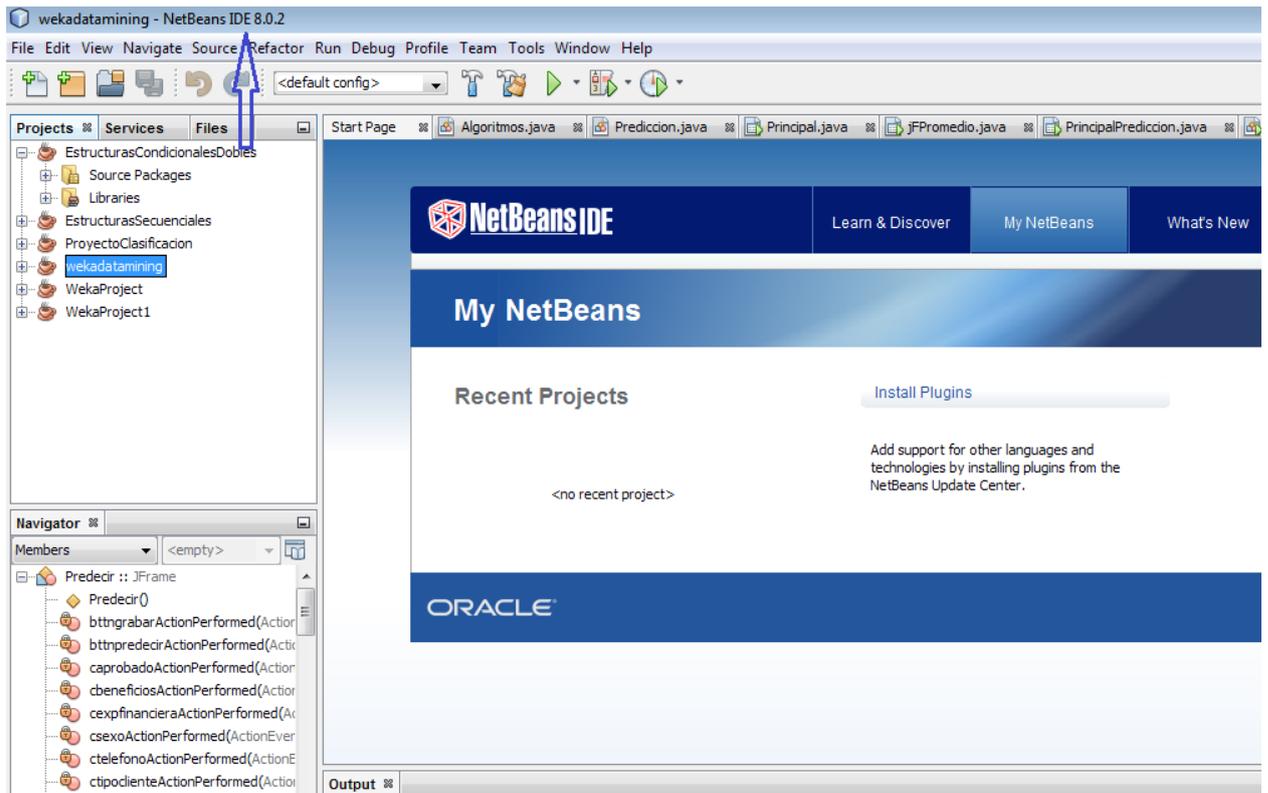


Figura 34. Versión de Java Netbeans.

Fuente: resultado de ejecución de Java Netbeans 8.0.2.

5.3. Conexión y manejo de procesos con Weka.

Primeramente vamos a realizar la creación de un JFrame (Figura 35) llamado *Principal* dentro del Source Packages de Java (Figura 36) llamado *Vistas*, creado inicialmente; ya que vamos a necesitar un entorno gráfico para manipular los resultados:

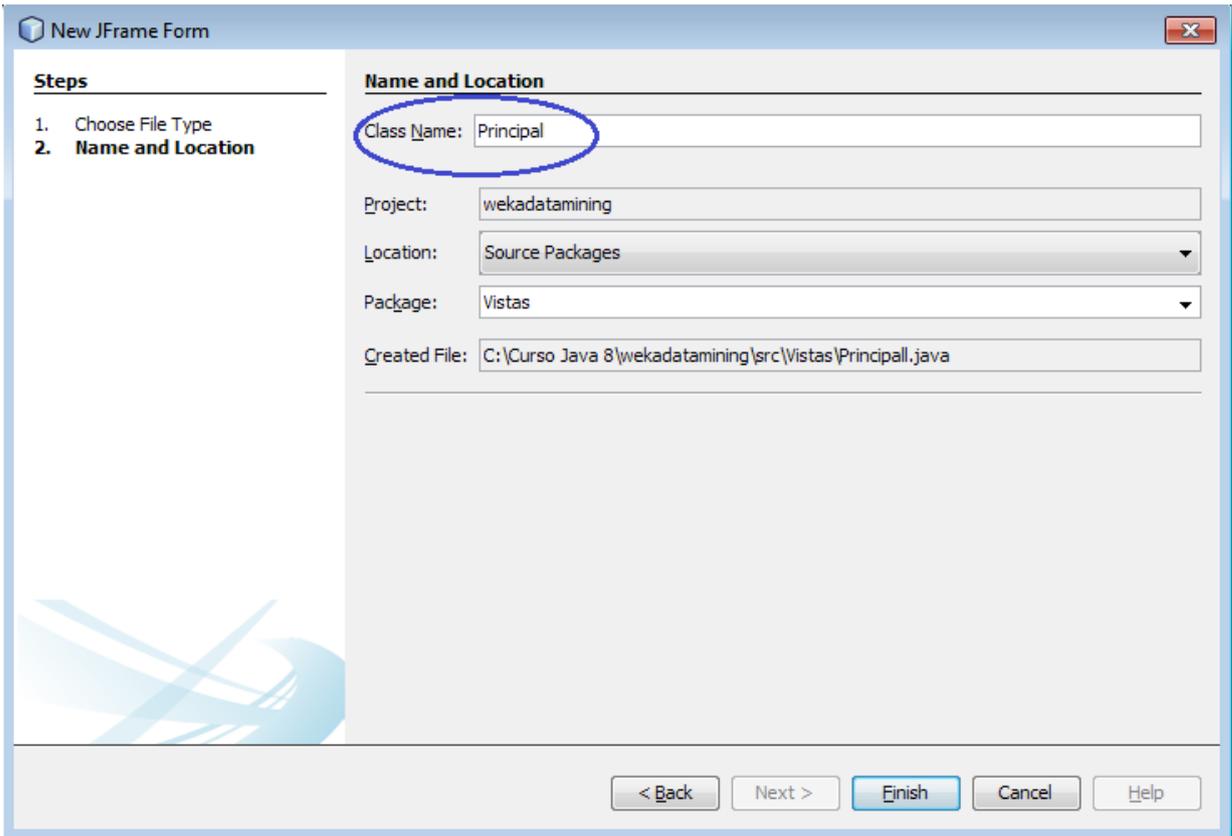


Figura 35. Creación de un JFrame.

Fuente: resultado de ejecución del proyecto de Java Netbeans 8.0.2.

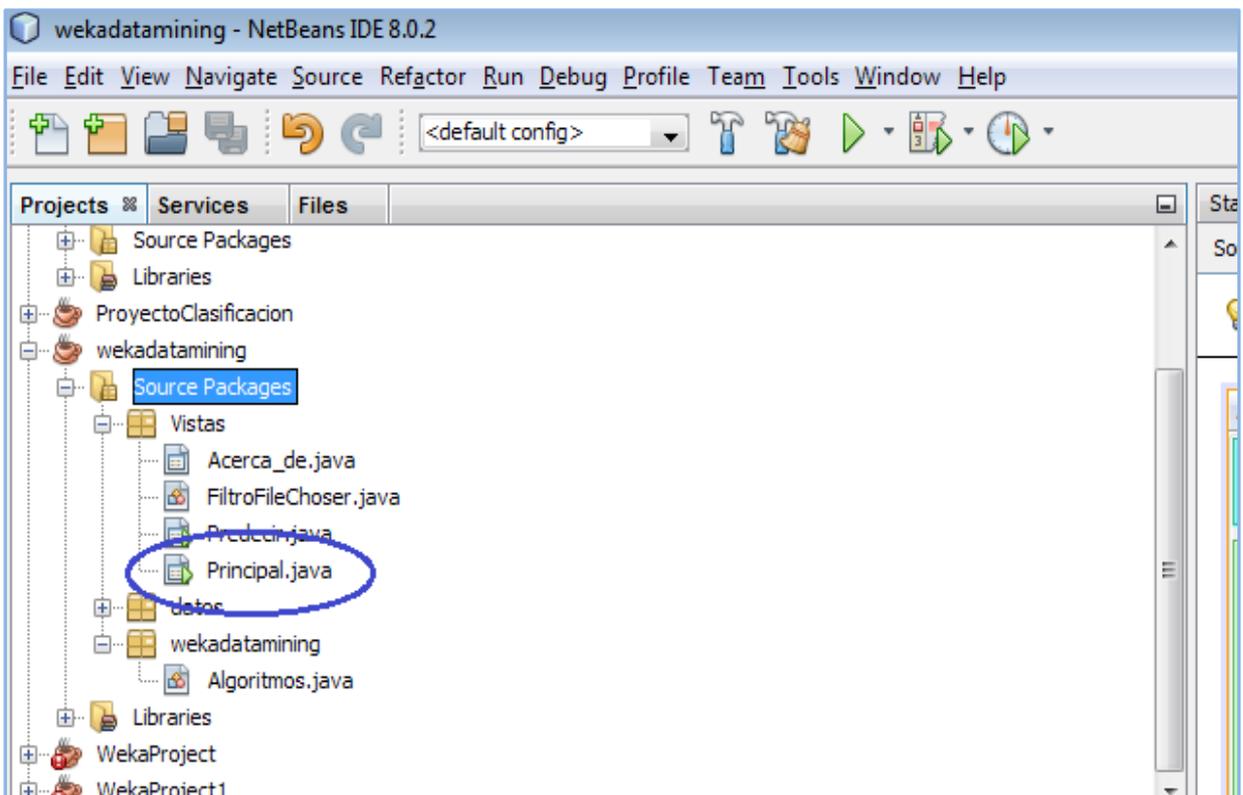


Figura 36. Source Packages de Java.

Fuente: resultado de ejecución del proyecto de Java Netbeans 8.0.2.

El JFrame principalmente va a utilizar la clase *Algoritmos* dentro de *Vistas*, creada dentro de la clase *wekadatamining* del proyecto; dicha clase es la que nos va a permitir conectarnos a los API¹⁶ de Weka y utilizar los algoritmos que se utilizan en dicha plataforma, pero para ello tenemos que agregar librerías específicas (Figura 37), por ejemplo *weka.jar* que se encuentra dentro del directorio de instalación de la plataforma en nuestro computador, en nuestro caso se encuentra en: *C:\Program Files\Weka-3-6\weka.jar*, por medio de *Add Jar/Folder*.

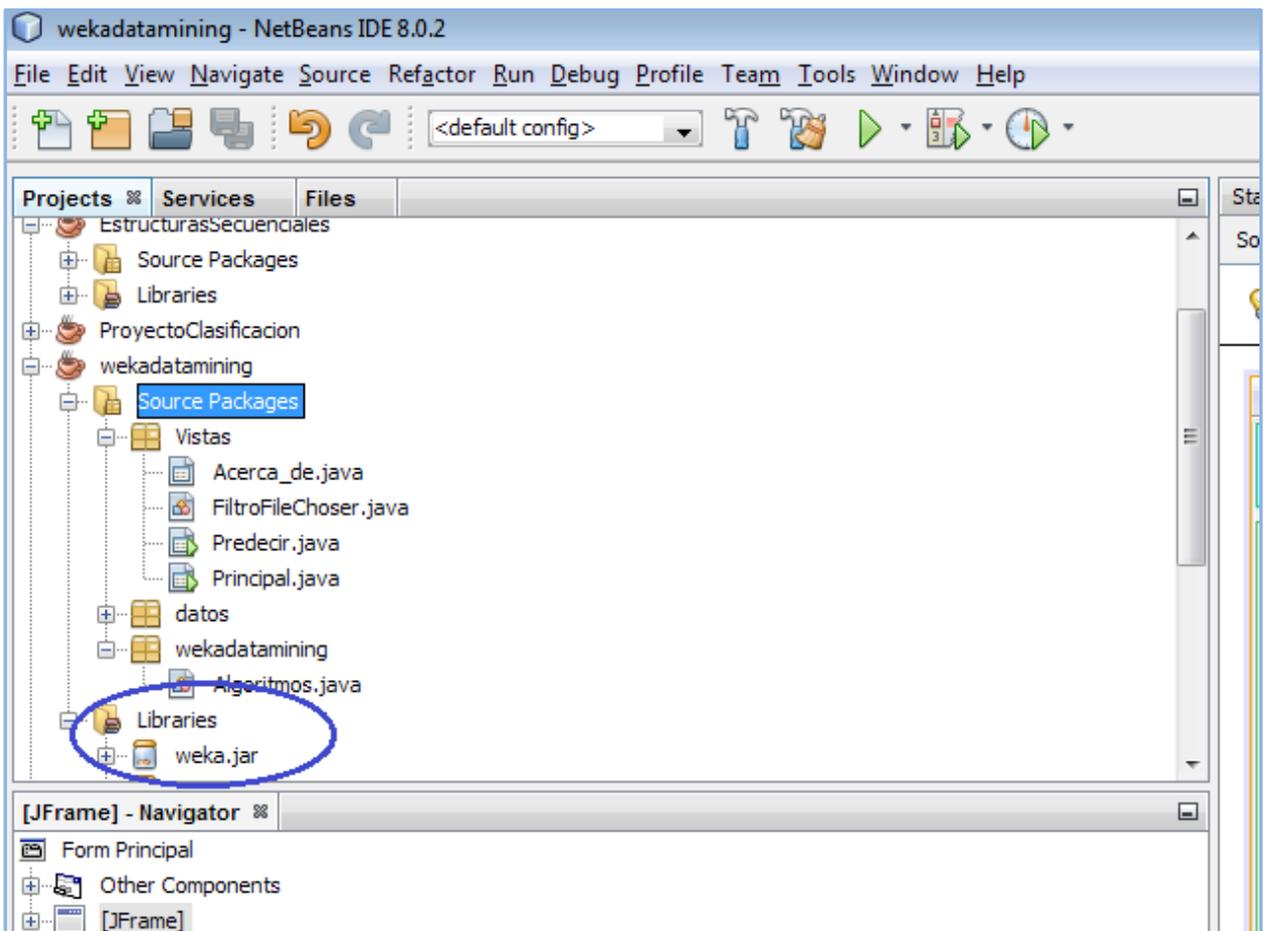


Figura 37. Agregar librerías específicas.

Fuente: resultado de ejecución del proyecto de Java Netbeans 8.0.2.

Al ejecutar el JFrame Principal, llama a la clase *Algoritmos*, y nos aparece la pantalla principal Data Mining (Figura 38) de la siguiente manera:

¹⁶ Significa Application Programming Interface, es el conjunto de subrutinas, funciones, métodos que pueden ser utilizados por otro software mediante bibliotecas.

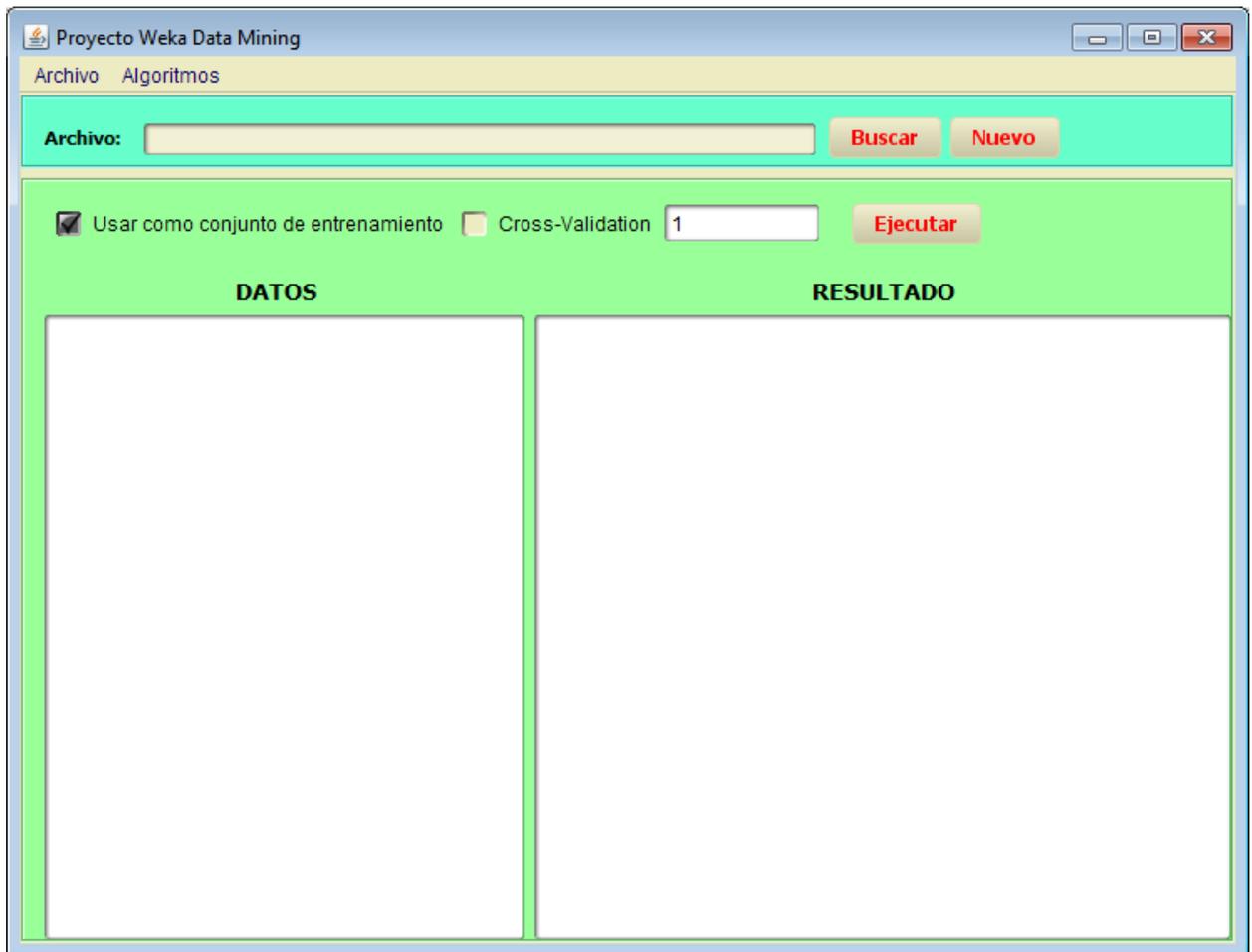


Figura 38. Pantalla principal Data Mining.

Fuente: resultado de ejecución del proyecto de Java Netbeans 8.0.2.

La anterior figura es la pantalla principal en donde vamos a realizar las siguientes tres operaciones:

- Selección del archivo .arff para la clasificación (Figura 39) por medio de Weka.
- Selección del algoritmo de clasificación (Figura 40) para realizar las pruebas.
- Presentación de resultados (Figura 41).

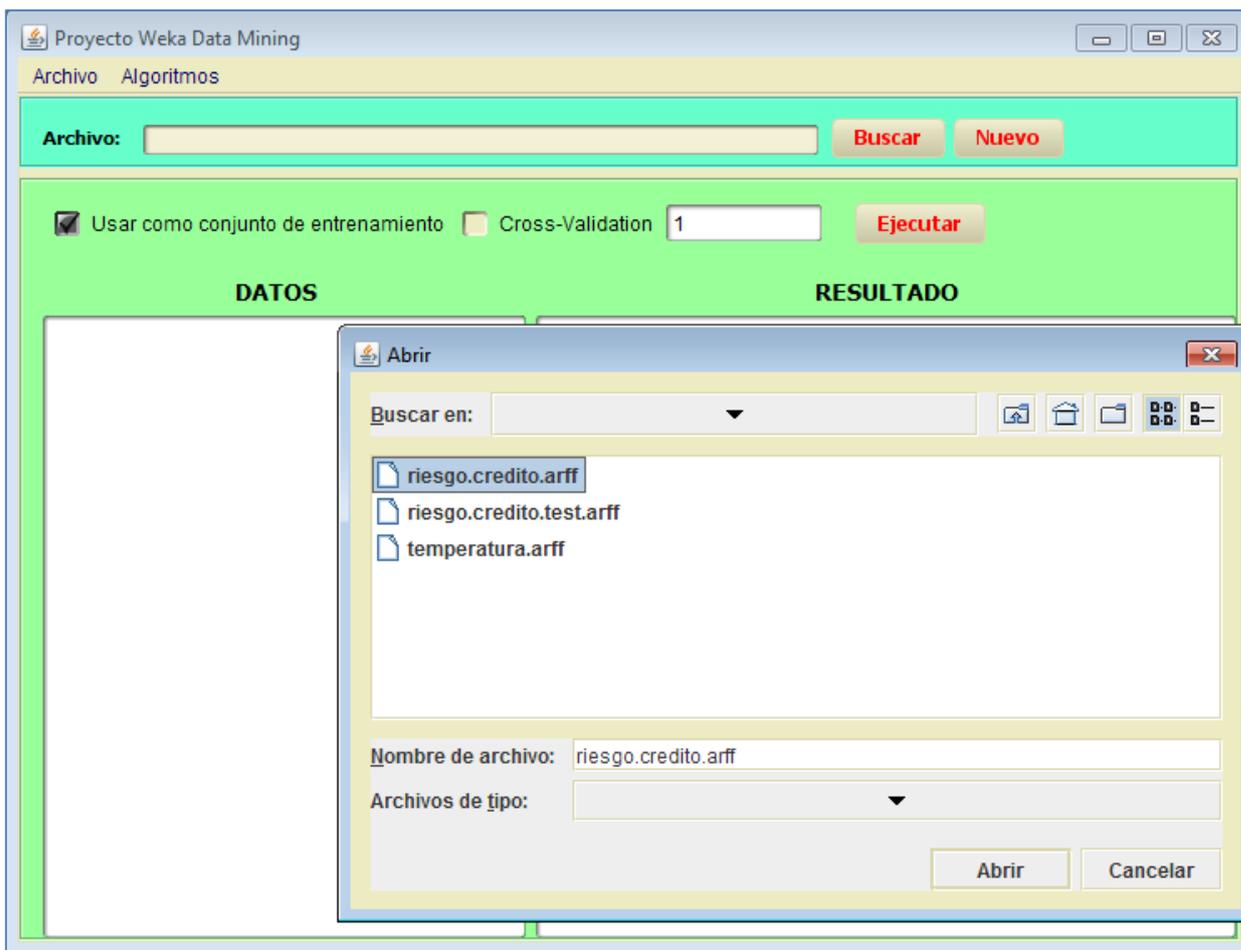


Figura 39. Selección del archivo .arff para la clasificación.

Fuente: resultado de ejecución del proyecto de Java Netbeans 8.0.2.

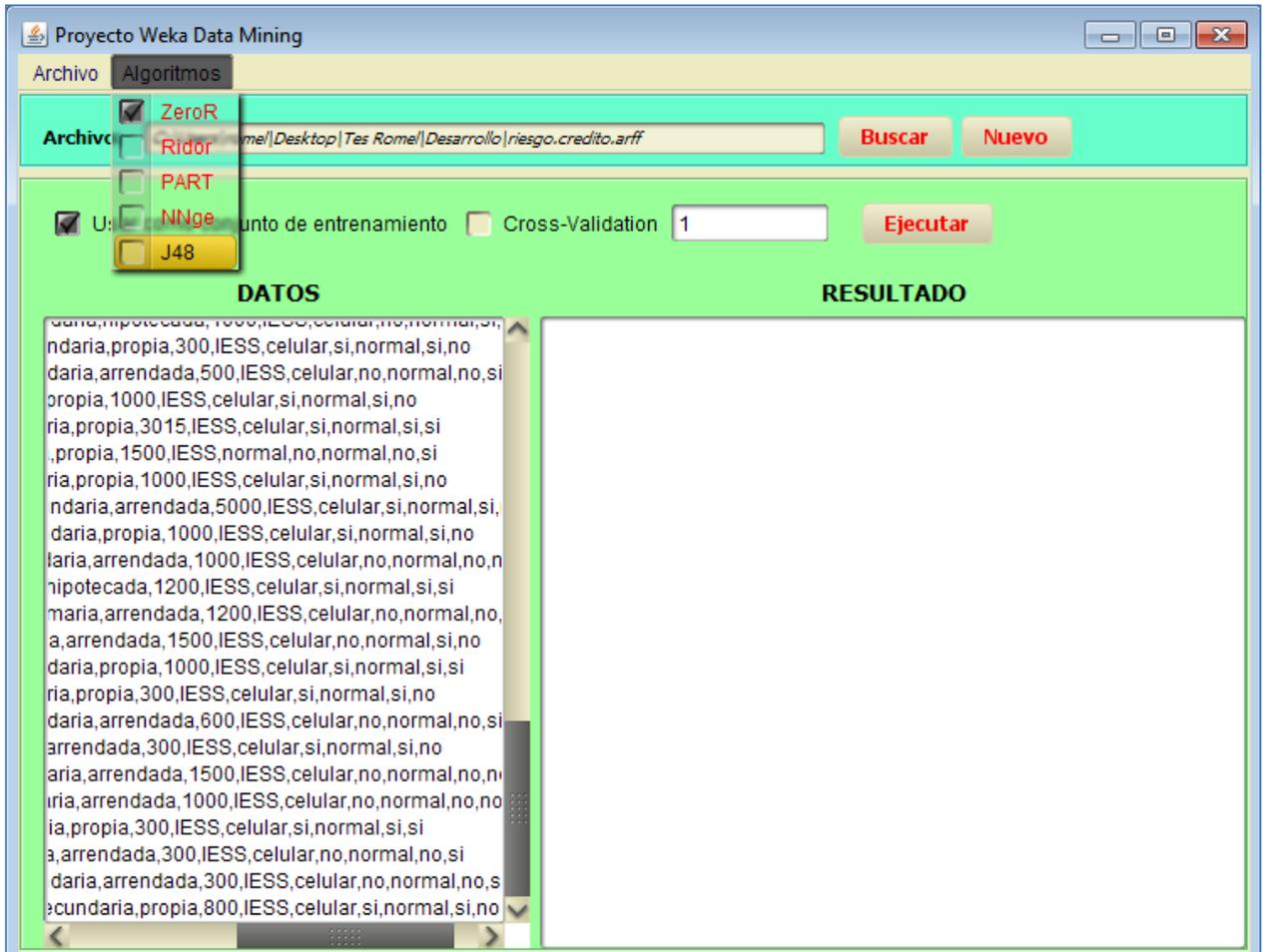


Figura 40. Selección del algoritmo de clasificación.

Fuente: resultado de ejecución del proyecto de Java Netbeans 8.0.2.

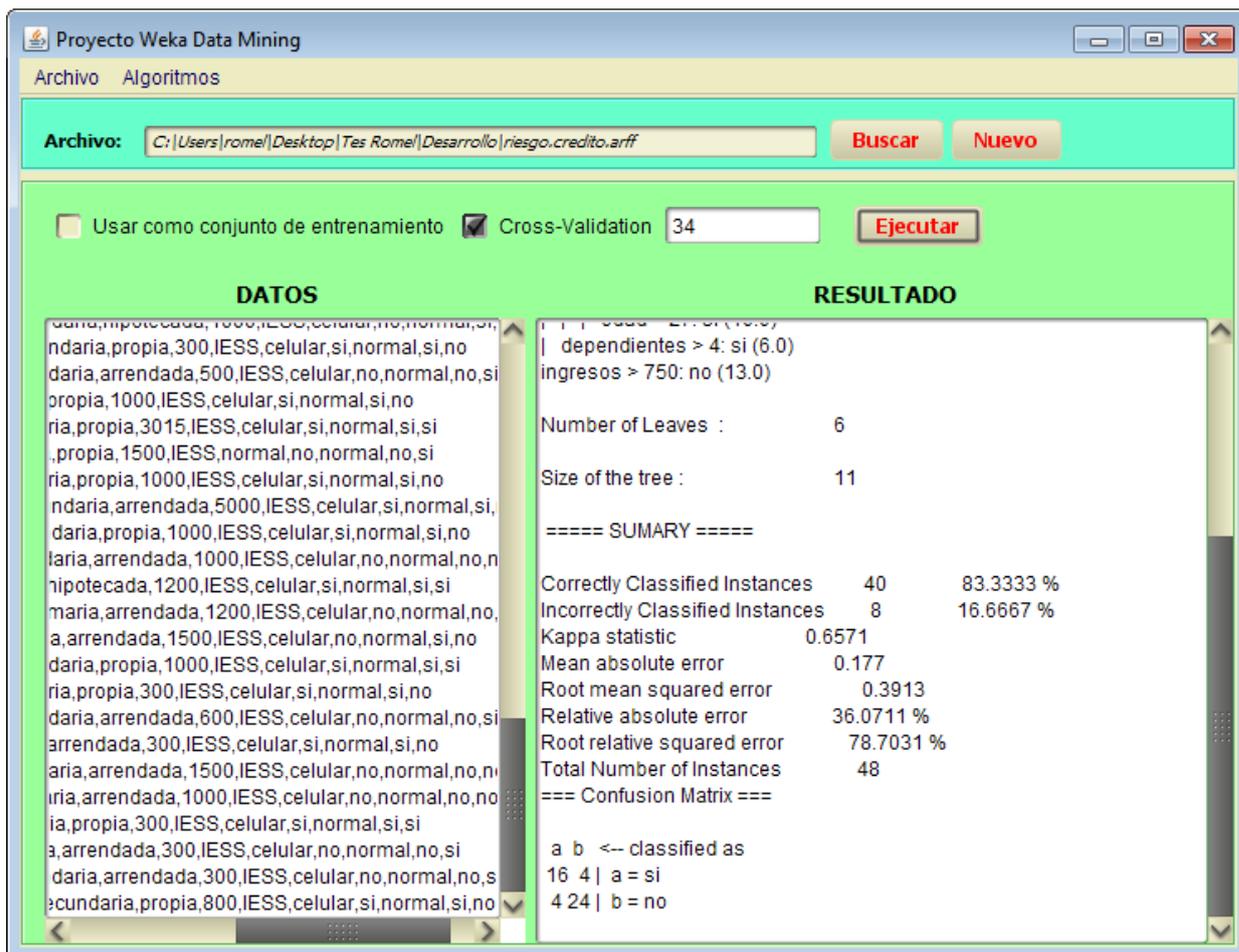


Figura 41. Presentación de resultados

Fuente: resultado de ejecución del proyecto de Java Netbeans 8.0.2.

Todo este proceso se trata de tomar el archivo .arff en donde tenemos los datos almacenados, los cuales vamos a procesar y nos servirá como entrenamiento.

5.4. Interfaz de usuario para presentación de resultados.

En esta sección vamos a utilizar los nuevos datos que el usuario va a utilizar para la predicción, es decir toma los datos de un nuevo cliente y los procesa para verificar si aplica o no a un crédito y poder tomar una decisión o comparar resultados; para la interfaz de usuario¹⁷ hemos creado una pantalla en donde el usuario ingresa la información, se realiza el proceso, se hace la comparación y después se guarda la información para su almacenamiento respectivo.

¹⁷ La interfaz de usuario es el medio con el que un usuario puede comunicarse con un equipo o un programa, los cuales son fáciles de entender y utilizar.

Para poder este fin hemos procedido en nuestro proyecto principal con la creación del JFrame *Predecir* (Figura 42):

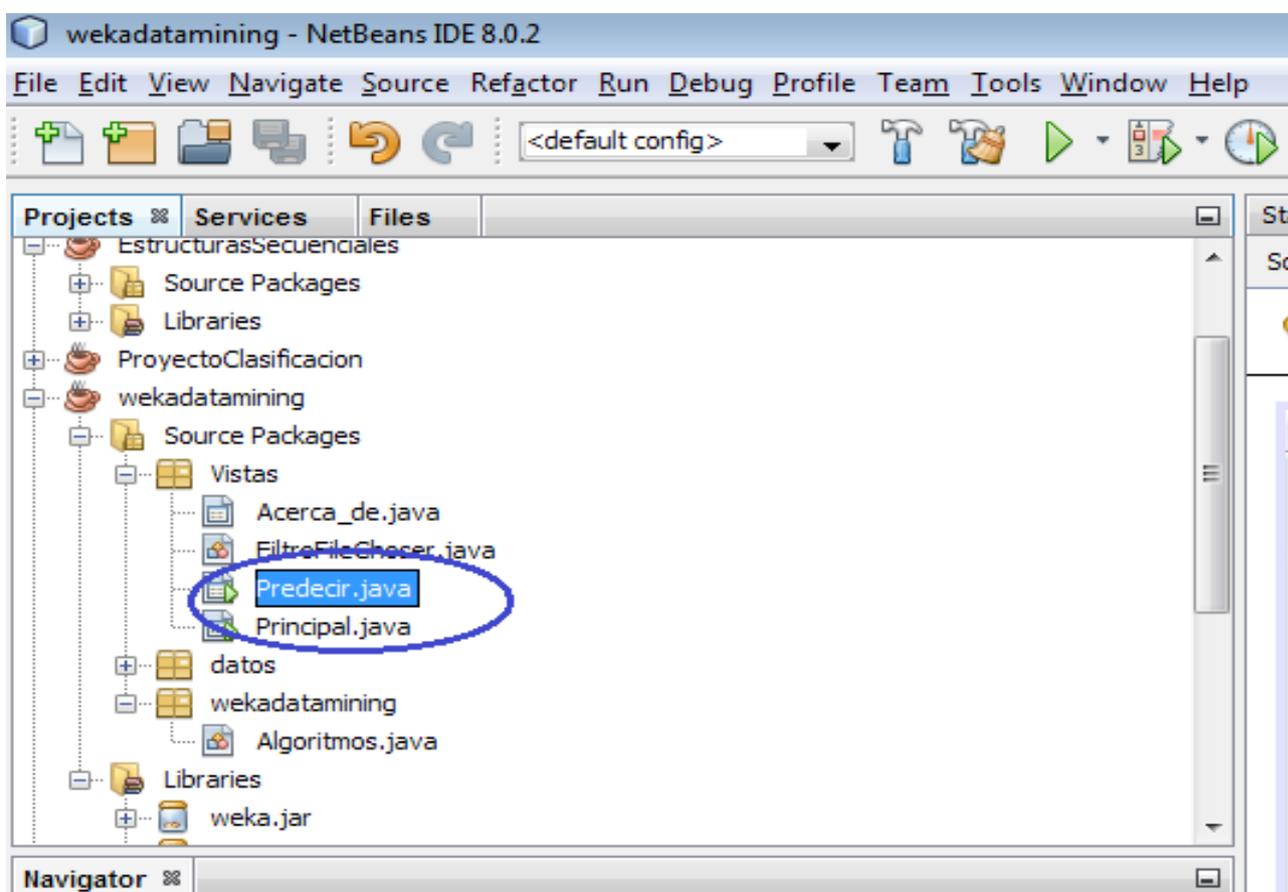


Figura 42. Creación del JFrame *Predecir*.

Fuente: resultado de ejecución del proyecto de Java Netbeans 8.0.2.

Al ejecutar *Predecir* (Figura 43) observamos que existe un campo nuevo llamado *aprobado* en vez de *mora* para almacenar la información para verificar si se aprueba o no un crédito a un cliente; el código fuente lo presentaremos en el Anexo 7:

Figura 43. Pantalla principal de ingreso de datos.

Fuente: resultado de ejecución del proyecto de Java Netbeans 8.0.2.

Según nuestra evaluación previa que se realizó en el análisis con Weka, el algoritmo que mejor nos dio resultados es el J48 con un cross validation de 18 y reduciendo los atributos a los que mejor se clasificaban, tales como : *edad*, *ingresos*, *dependientes*, *vehículo*, *tipo.cliente* y *monto*, con lo cual verificamos las mejores variables o tendencias para determinar cuando un cliente está en **MORA**; entonces el algoritmo realiza una clasificación de estos atributos relevantes (Figura 44), tal como se muestra en la siguiente pantalla en el árbol del J48:

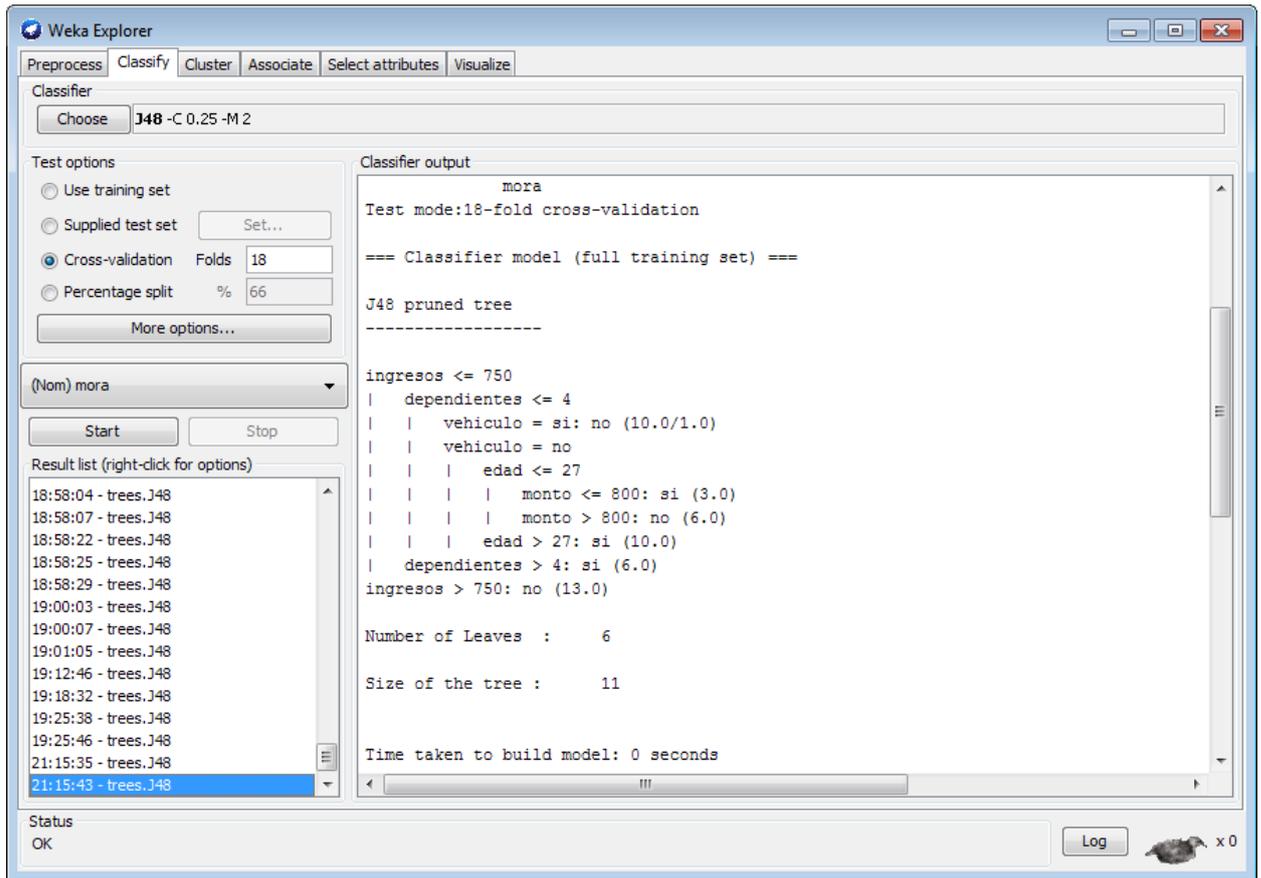
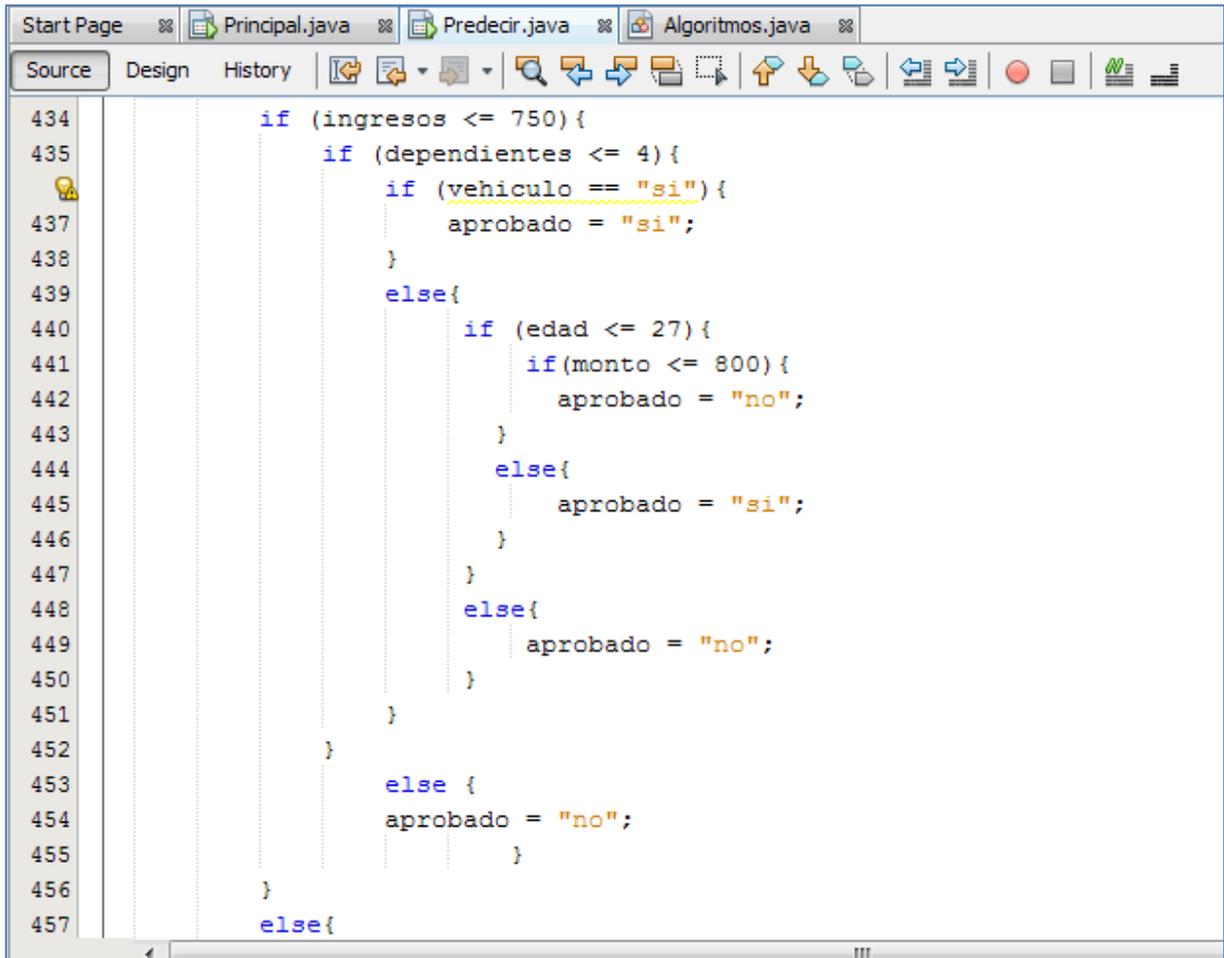


Figura 44. Clasificación de los atributos relevantes.

Fuente: resultado de ejecución del proyecto de Java Netbeans 8.0.2.

Dicho proceso realizado es el mismo dentro de la Programación del JFrame, en donde podemos ver el código fuente de *Predecir* (Figura 45):



```
434         if (ingresos <= 750) {
435             if (dependientes <= 4) {
436                 if (vehiculo == "si") {
437                     aprobado = "si";
438                 }
439             } else {
440                 if (edad <= 27) {
441                     if (monto <= 800) {
442                         aprobado = "no";
443                     }
444                 } else {
445                     aprobado = "si";
446                 }
447             } else {
448                 aprobado = "no";
449             }
450         }
451     }
452 }
453 else {
454     aprobado = "no";
455 }
456 }
457 else {
```

Figura 45. Código fuente de Predecir.

Fuente: resultado de ejecución del proyecto de Java Netbeans 8.0.2.

De ésta forma a servir de gran ayuda para la evaluación del resultado final al verificar si un cliente aprueba o no un crédito dentro de la cooperativa.

Al momento de procesar la información en la pantalla de los resultados, hay que señalar que solamente se va a ejecutar el proceso de los atributos más relevantes (Figura 46) como son **ingresos**, **dependientes**, **vehículo**, **edad**, **monto** los otros datos se ingresan por cuestiones de almacenamiento por parte del usuario; vamos a realizar una demostración para ilustrar este proceso, ingresando los datos de un nuevo cliente el cual ha solicitado un préstamo a la cooperativa:

sexo = masculino, edad = 33, ingresos = 1200, dependientes = 3, estado civil = casado, instrucción = superior, actividad = secundaria, vivienda = arrendada, monto =

2500, beneficios = iess, teléfono = celular, vehiculo = si, tipo de cliente = normal, experiencia financiera = no, aprobado = no.

Estos datos van a ser ingresados por el usuario en la pantalla, los cuales se manejan en la entidad para evaluar a los clientes y la decisión inicial que ha tomado el personal de crédito, es de **NO** aprobar el crédito.

Sexo:	<input type="text" value="masculino"/>	Instrucci...:	<input type="text" value="superior"/>	Telefono:	<input type="text" value="celular"/>
Edad:	<input type="text" value="33"/>	Actividad:	<input type="text" value="secundaria"/>	Vehiculo:	<input type="text" value="si"/>
Ingresos:	<input type="text" value="1200"/>	Vivienda:	<input type="text" value="arrendada"/>	Tipo Cliente:	<input type="text" value="normal"/>
Dependiente...:	<input type="text" value="3"/>	Monto:	<input type="text" value="2500"/>	Exp Financi...:	<input type="text" value="no"/>
Estado Civil:	<input type="text" value="casado"/>	Benefici...:	<input type="text" value="iess"/>	Aprobado:	<input type="text" value="no"/>

Limpiar Grabar Salir Regresar Inicio

PREDECIR

Figura 46. Proceso de los atributos más relevantes.

Fuente: resultado de ejecución del proyecto de Java Netbeans 8.0.2.

Como podemos observar en la pantalla, al ejecutar el botón PREDECIR el programa se va a realizar la predicción de acuerdo a los análisis que nosotros hicimos anteriormente y por lo tanto la sugerencia del programa de acuerdo a la predicción (Figura 47), es **SI** con lo cual se obtendrá una valiosa ayuda para la toma de decisiones, con lo cual el oficial de crédito podrá evaluar nuestro resultado y tomar las medidas pertinentes:

COOPERATIVA DE AHORRO Y CREDITO - PREDICCIÓN DE RIESGOS

Sexo:	masculino	Instrucci...:	superior	Telefono:	celular
Edad:	33	Actividad:	secundaria	Vehiculo:	si
Ingresos:	1200	Vivienda:	arrendada	Tipo Cliente:	normal
Dependiente...:	3	Monto:	2500	Exp Financi...:	no
Estado Civil:	casado	Benefici...:	iess	Aprobado:	no

Limpiar Grabar Salir Regresar Inicio

PREDECIR

RESOLUCION: no SUGERENCIA: SI

Figura 47. Sugerencia del programa de acuerdo a la predicción.

Fuente: resultado de ejecución del proyecto de Java Netbeans 8.0.2.

Con el botón de Grabar, la información se graba en un archivo de Excel para que el usuario pueda realizar un registro de los datos (Figura 48):

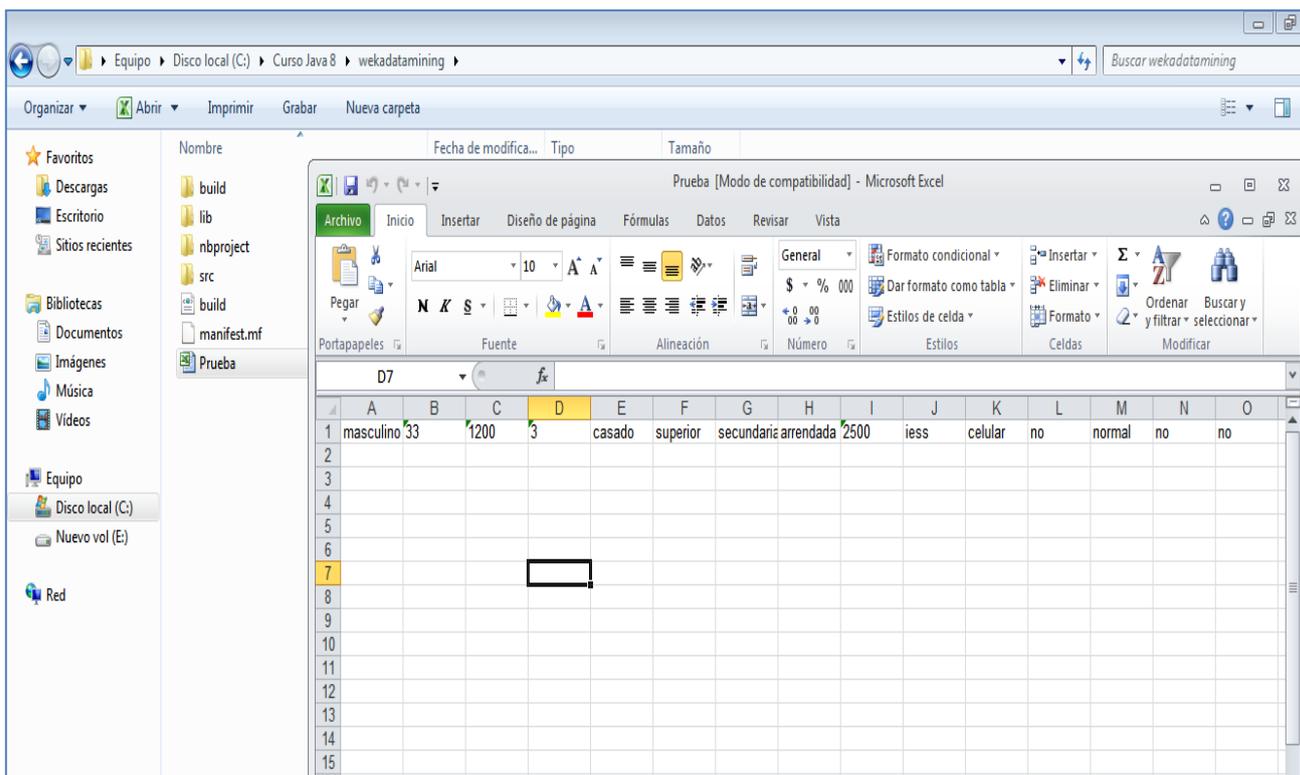


Figura 48. Registro de los datos.

Fuente: resultado de ejecución del proyecto de Java Netbeans 8.0.2.

Por último al presionar Limpiar borra los datos de las celdas y permite ingresar nueva información, y con Salir cerramos la pantalla.

5.5. Aplicación del modelo predictivo desarrollado.

El modelo predictivo desarrollado con su programa entregable, una vez finalizado se procede a ser instalado en la oficina de la cooperativa de Ahorro y Crédito Nuevo Amanecer CACNA; dicha institución es de clase pequeña, de reciente creación y dedicada al microcrédito. El programa desarrollado puede aplicarse sin problemas en el ámbito de éste tipo de instituciones financieras, ya sean pequeñas, medianas o grandes, en donde no se dispone de programas sofisticados para la evaluación de riesgos crediticios, ya que por lo general dentro de las mismas se trabaja con los mismos parámetros y variables para evaluar y realizar el otorgamiento de crédito a los clientes.

Se ha creado un ejecutable del aplicativo (Figura 49) realizado en Java para la ejecución del programa, al cual le hemos denominado CACNA, como el nombre de la institución financiera. El personal de la entidad financiera que se encarga de receptor las peticiones de los clientes y de realizar la evaluación de los mismos, podrá utilizar la aplicación previa

capacitación de parte nuestra; para ello se realizará un manual de usuario para que pueda utilizar la aplicación sin ningún problema.



Figura 49. Icono ejecutable del programa desarrollado.

Fuente: creación del ejecutable del programa desarrollado.

El ejecutable nos permite abrir el aplicativo y verificar su funcionamiento (Figuras 50, 51) en el computador del usuario; previamente al personal encargado de utilizarlo se le instruyó acerca del funcionamiento del programa y podrá tomar decisiones sobre el otorgamiento de créditos con ayuda de un proceso técnico de ésta valiosa herramienta informática; igualmente se mantendrá un contacto inicial con el usuario por cualquier duda acerca del funcionamiento.

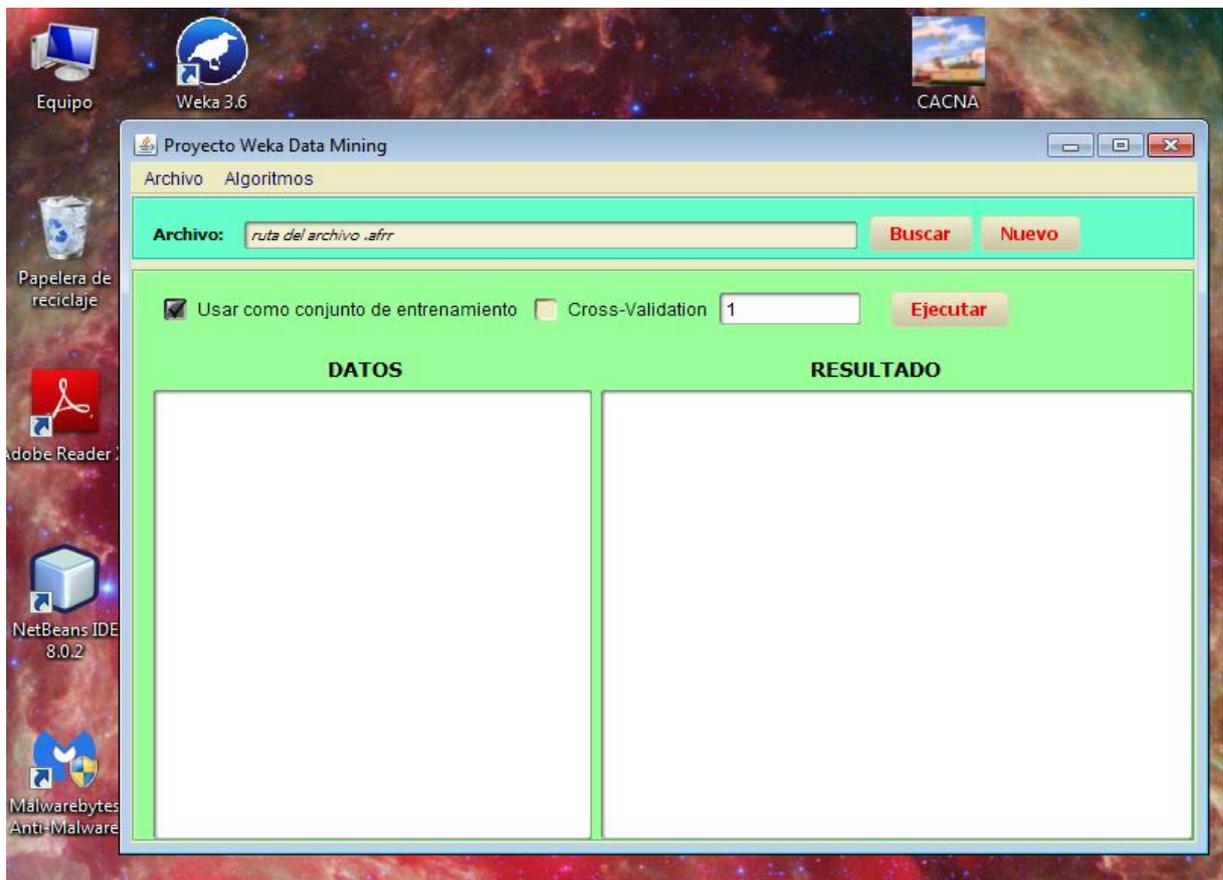


Figura 50. Verificación de funcionamiento en el computador del usuario.

Fuente: ejecución del aplicativo del modelo de riesgo crediticio.

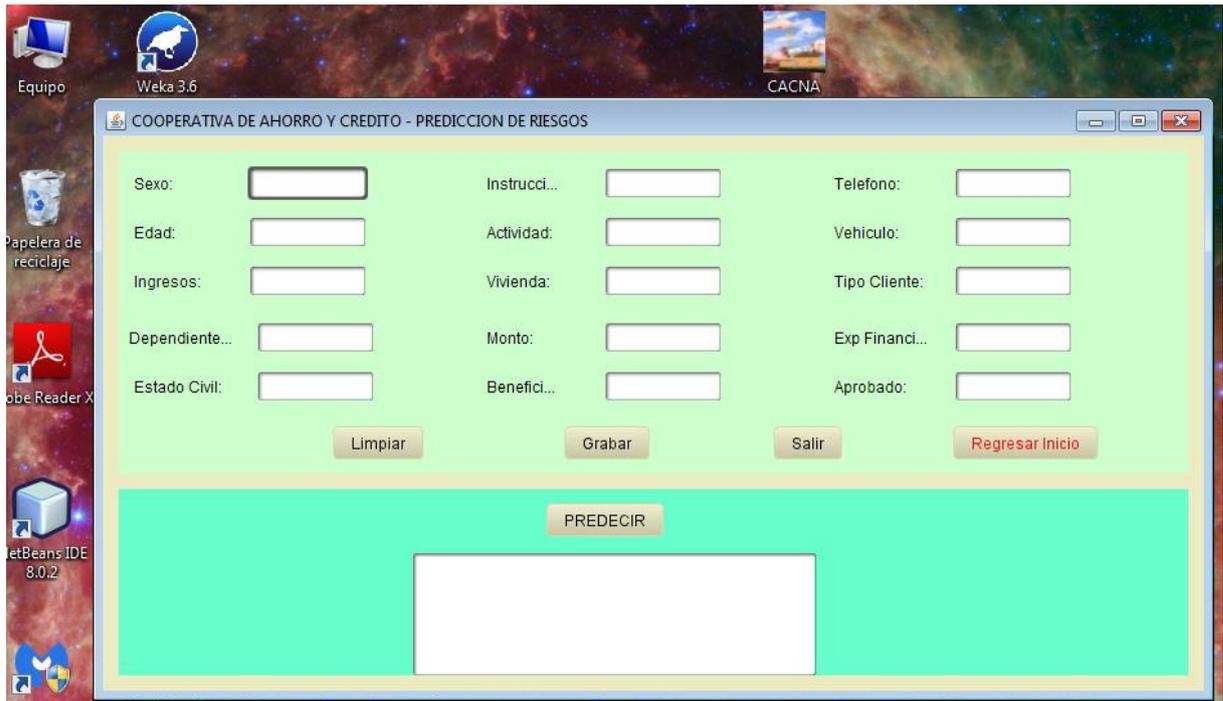


Figura 51. Verificación de funcionamiento en el computador del usuario.

Fuente: ejecución del aplicativo del modelo de riesgo crediticio.

CONCLUSIONES

- La Inteligencia Artificial resultó ser una buena herramienta para realizar clasificación de datos y realizar predicciones.
- La plataforma Weka que utilizamos en nuestro proyecto resultó ser una herramienta de valiosa ayuda, además de ser gratuita y sin restricciones de uso.
- Entre los diferentes algoritmos de clasificación que pudimos comprobar para el tratamiento de los datos el J48 resultó ser el que nos dio mejores resultados.
- Los métodos automatizados de predicción de datos unidos con la experiencia personal pueden dar óptimos resultados.

RECOMENDACIONES.

- El modelo implementado en nuestro proyecto no pretende sustituir a los procedimientos que actualmente se llevan a cabo en la cooperativa de ahorro y crédito, pero si se cuenta con una herramienta eficaz para la ayuda de toma de decisiones.
- La entidad en donde se desarrolló nuestro trabajo es pequeña, pero puede ser implementado en instituciones medianas y grandes, ya que por lo general se manejan parámetros similares de catalogación de clientes.
- Se podrían emitir implementar en el programa otras variables significativas de evaluación de créditos y así poder verificar con el personal a cargo si los resultados son los esperados.

BIBLIOGRAFIA

- Wikipedia. (2015). Inteligencia Artificial. Recuperado de http://es.wikipedia.org/wiki/Inteligencia_artificial
- Data Mining. (2015). Recuperado de <https://sistemas2009unl.wordpress.com/data-mining-para-agronomia/>
- Deitel, Harvey M. y Deitel, Paul J. (2004). Java, Como Programar, Quinta Edición. México: Prentice Hall.
- Ricardo Aler. (2009). Tutorial Weka 3.6.0. Recuperado de <http://ocw.uc3m.es/ingenieria-informatica/herramientas-de-la-inteligencia-artificial/contenidos/transparencias/TutorialWeka.pdf>.
- Monografías.com: computación. (2015). Inteligencia Artificial. Recuperado de <http://www.monografias.com/trabajos16/la-inteligencia-artificial/la-inteligencia-artificial.shtml>.
- Wikipedia. (2015). Minería de Datos. Recuperado de http://es.wikipedia.org/wiki/Miner%C3%ADa_de_datos.
- Weka the University of Waikato. (2015). Weka 3: Data Mining Software in Java. Recuperado de <http://www.cs.waikato.ac.nz/ml/weka/>
- Weka the University of Waikato. (2015). Documentation. Recuperado de <http://www.cs.waikato.ac.nz/ml/weka/documentation.html>.
- Superintendencia de Bancos del Ecuador. (2015). Riesgo Crediticio. Recuperado de http://www.sbs.gob.ec/practg/p_index?vp_art_id=1&vp_tip=11&vp_lang=1&vp_buscr=1.
- Monografías.com. (2015). Riesgo Crediticio. Recuperado de <http://www.monografias.com/trabajos-pdf/riesgo-crediticio/riesgo-crediticio.pdf>.
- My Credit Union. (2015). Concepto de cooperativa de ahorro y crédito. Recuperado de <http://espanol.mycreditunion.gov/about-credit-unions/Pages/what-is-a-credit-union.aspx>.
- Aplicaciones empresariales de Data Mining. (2015). Recuperado de <http://www.idescat.cat/sort/questiio/questiio/pdf/25.3.5.Garrido.pdf>.
- Minería de Datos. (2015). Recuperado de <http://www.monografias.com/trabajos56/mineria-de-datos-venezuela/mineria-de-datos-venezuela2.shtml>.
- Performance Comparison of Machine Learning Algorithms on Integration of Clustering and Classification Techniques. (2015). Recuperado de <http://iasir.net/IJETCASpapers/IJETCAS13-144.pdf>.

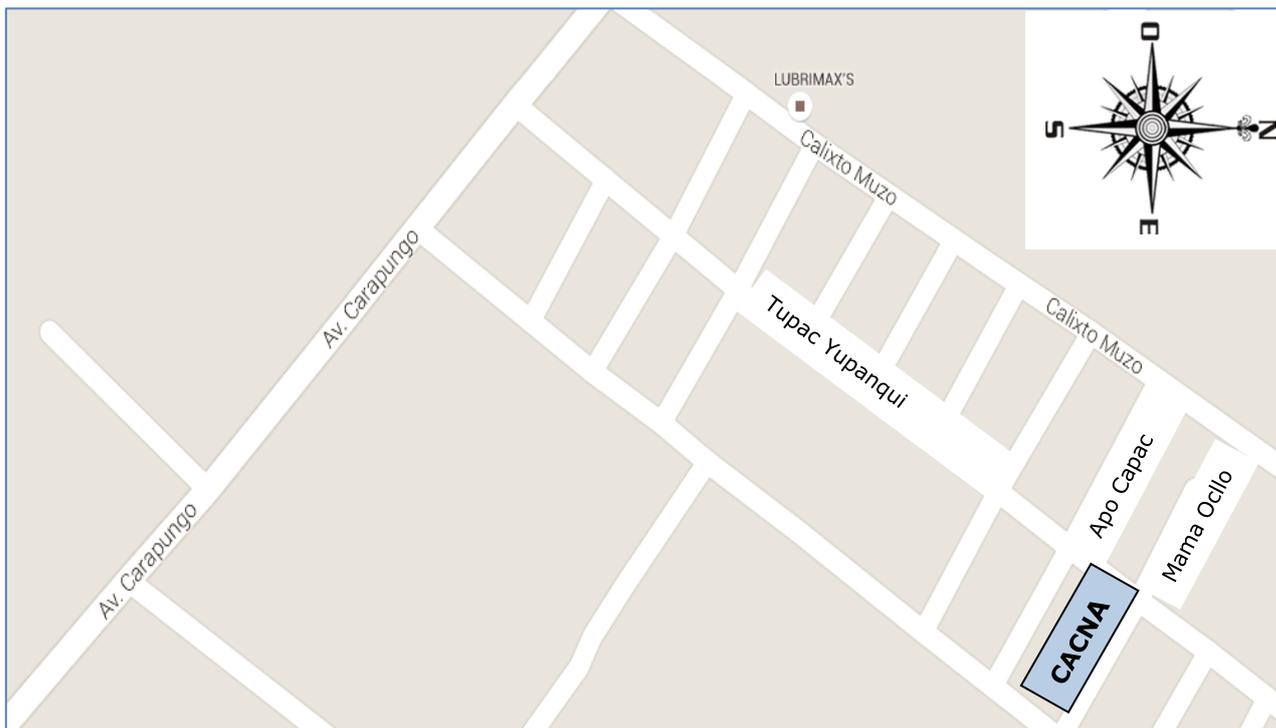
ANEXOS

ANEXO 1

Fotos del local de la cooperativa CACNA y su ubicación dentro de Quito



La cooperativa de ahorro y crédito se encuentra ubicada en el norte de la ciudad de Quito, por el sector de Llano Grande.



ANEXO 2
Formulario de Solicitud de Crédito

 COOP. DE AHORRO Y CREDITO "NUEVO AMANECER". Ltda SOLICITUD DE CREDITO				TITULAR GARANTE	
OFICINA.....		OFICIAL.....		FECHA DE INGRESO SISTEMA.....	
CREDITO No.....		SOLICITUD No.....		TIPO DE CREDITO.....	
MONTO SOLICITADO: \$.....		MONTO APROBADO: \$.....		PARA INVERTIR EN:.....	
PUEDO PAGAR EL VALOR DE \$.....		SEMANAL <input type="checkbox"/> QUINCENAL <input type="checkbox"/> MENSUAL <input type="checkbox"/>		\$..... TASA DE PAGO.....	
NUMERO DE CUENTA DE AHORROS.....		COORDENADAS DE APORTACION.....		APELLIDOS Y NOMBRES DEL GARANTE (S) 1..... 2.....	
DATOS GENERALES DEL SOLICITANTE/GARANTE					
APELLIDO PATERNO.....		APELLIDO MATERNO.....		NOMBRES COMPLETOS.....	
NACIONALIDAD ECUATORIANA <input type="checkbox"/> OTRA.....		No. DE CEDULA/PASAPORTE.....		LUGAR DE NACIMIENTO.....	
FECHA DE NACIMIENTO.....		EDAD.....			
ESTADO CIVIL SOLTERO <input type="checkbox"/> UNION LIBRE <input type="checkbox"/> CASADO <input type="checkbox"/> VIUDO <input type="checkbox"/> DIVORCADO <input type="checkbox"/>		NUMERO DE CARGAS.....		NIVEL DE EDUCACION PRIMARIA <input type="checkbox"/> POSTGRADO <input type="checkbox"/> SECUNDARIA <input type="checkbox"/> OTRO <input type="checkbox"/> SUPERIOR <input type="checkbox"/>	
PROFESION/OFICIO.....		ACTIVIDAD.....			
DIRECCION DOMICILIARIA DEL SOLICITANTE/GARANTE					
PROVINCIA.....		CANTON.....		CIUDAD.....	
PARROQUIA.....		BARRIO.....			
AVENIDA CALLE PRINCIPAL/MANZANA/ETAPA/SECTOR/LOTE.....		No. CALLE SECUNDARIA/TRANSVERSAL.....		REFERENCIA UBICACION (Parques, Iglesia, Escuela).....	
VIVIENDA <input type="checkbox"/> PROPIA <input type="checkbox"/> FAMILIAR <input type="checkbox"/> <input type="checkbox"/> ARRENDADA <input type="checkbox"/> ANTICIPES		TELEFONO.....		CELULAR.....	
E-MAIL.....				Si su actividad es menor a 1 año indique domicilio anterior.....	
DESCRIPCION DE LA ACTIVIDAD					
EMPLEADO <input type="checkbox"/> INDEPENDIENTE <input type="checkbox"/> MICROEMPRES. <input type="checkbox"/> RENTISTA <input type="checkbox"/> JUBILADO <input type="checkbox"/> ESTUDIANTE <input type="checkbox"/> OTRO.....					
EMPRESA DONDE TRABAJA EL SOLICITANTE/GARANTE					
NOMBRE DE LA EMPRESA/NEGOCIO.....		CARGO QUE DESEMPEÑA.....		SUELDO O INGRESO FIJO <input type="checkbox"/> VARIABLE <input type="checkbox"/>	
PROVINCIA DE UBICACION EMPRESA/NEGOCIO.....		CANTON.....		CIUDAD.....	
PARROQUIA.....		BARRIO.....			
AVENIDA CALLE PRINCIPAL/MANZANA/ETAPA/SECTOR/LOTE.....		No. CALLE SECUNDARIA/TRANSVERSAL.....		REFERENCIA UBICACION.....	
TELEFONO.....		CELULAR.....			
ACTIVIDAD ECONOMICA/NEGOCIO					
COMERCIO <input type="checkbox"/> INDUSTRIAL <input type="checkbox"/> CONSTRUCCION <input type="checkbox"/> SERVICIO <input type="checkbox"/>					
TIEMPO DE TRABAJO AÑOS.....		MESES.....		Si su tiempo de trabajo es menor a 1 año indique trabajo anterior.....	
DATOS GENERALES DEL CONYUGUE SOLICITANTE/GARANTE					
APELLIDO PATERNO.....		APELLIDO MATERNO.....		NOMBRES COMPLETOS.....	
NACIONALIDAD ECUATORIANA <input type="checkbox"/> OTRA.....		No. DE CEDULA/PASAPORTE.....		LUGAR DE NACIMIENTO.....	
FECHA DE NACIMIENTO.....		EDAD.....			
DESCRIPCION DE LA ACTIVIDAD					
EMPLEADO <input type="checkbox"/> INDEPENDIENTE <input type="checkbox"/> MICROEMPRES. <input type="checkbox"/> RENTISTA <input type="checkbox"/> JUBILADO <input type="checkbox"/> ESTUDIANTE <input type="checkbox"/> OTRO.....					
NOMBRE DE LA EMPRESA/NEGOCIO.....		TIEMPO DE TRABAJO AÑOS.....		DIRECCION.....	
SUELDO \$.....					
ACTIVIDAD DE LA EMPRESA /NEGOCIO					
TELEFONO.....		TIEMPO DE TRABAJO AÑOS.....		MESES.....	
				Si su tiempo de trabajo es menor a 1 año indique trabajo anterior.....	
FAMILIAR QUE NO VIVA CON USTED					
APELLIDOS Y NOMBRES.....		C.I.....		DIRECCION DE DOMICILIO.....	
FAMILIAR.....		TELEFONOS.....		HORAS DE CONTACTO.....	
REFERENCIAS PERSONALES					
APELLIDOS Y NOMBRES.....		DIRECCION DE DOMICILIO/TRABAJO.....		TELEFONOS.....	
1.....					
2.....					
REFERENCIAS BANCARIAS					
ENTIDAD.....		(AHORROS, CTA, CTE, PAQUETA).....		NUMERO DE CUENTA, FARETA.....	
1.....				CUPO \$.....	
2.....				\$.....	
3.....				\$.....	
REFERENCIAS COMERCIALES					
APELLIDOS Y NOMBRES.....		DIRECCION DE DOMICILIO/TRABAJO.....		TELEFONOS.....	
1.....					
2.....					

ANEXO 3

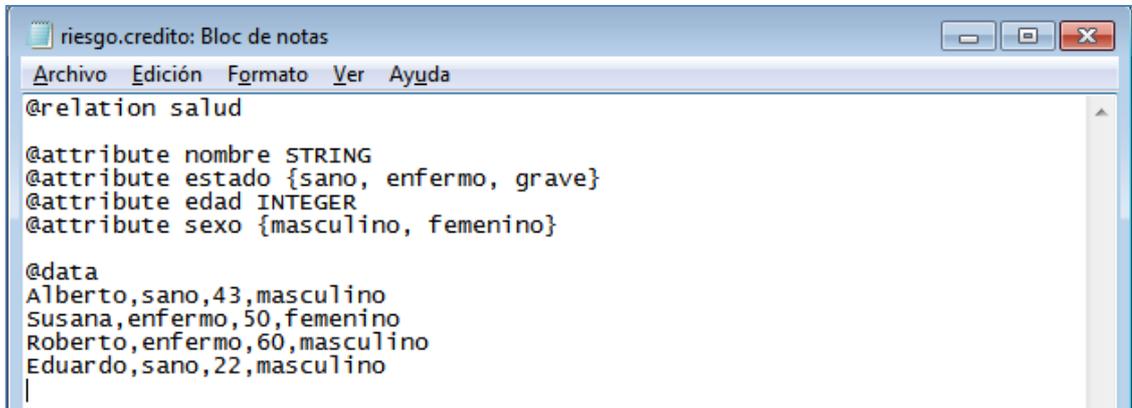
Como crear un archivo *.arff*

Un archivo tipo *.arff* para poder procesarlo en weka consta de las siguientes partes:

- **Cabecera:** en esta parte se define el nombre de la relación o el tema a tratar:
`@relation <nombre-de-la-relación>`
En donde `<nombre-de-la-relación>` es de tipo string y si contiene algún espacio tiene que ir entre comillas.
- **Declaración de atributos:** en donde constan todos los atributos que se van a utilizar en el archivo; se declaran de la siguiente manera:
`@attribute <nombre-del-atributo> <tipo>`
Igual que el caso anterior `< nombre-del-atributo>` es de tipo string y si contiene algún espacio tiene que ir entre comillas. Weka soporta los siguientes tipos de datos:
 - **String:** para denominar cadenas de texto
 - **Integer:** para denominar números enteros
 - **Numeric:** para expresar números reales
 - **Date:** se utiliza para fechas mediante un formato entre comillas y caracteres separadores y unidades de tiempo:
 - dd para Día
 - MM para Mes
 - yyyy representa Año
 - HH significa Horas
 - mm para los Minutos
 - ss representa los Segundos
 - **Enumerado:** aquí se determinan posibles valores que puede tomar el atributo representados en caracteres. Por ejemplo para definir el estado de salud de una persona:
`@attribute estado {sano, enfermo, grave}`
- **Sección de Datos:** En esta parte definimos los datos que se van a utilizar en la relación, separados los valores de los atributos con comas y las relaciones en la siguiente línea. Se representa de la siguiente forma:
`@data`
José, Sano, 43, masculino

Una vez que tenemos definido las elementos ya partes de nuestro archivo *.arff* podemos utilizar los siguientes dos métodos para su creación:

1. Escribimos todos los datos del archivo, es decir la cabecera y el data en un block de notas y lo guardamos con la extensión *.arff*: *riesgo.credito.arff*:

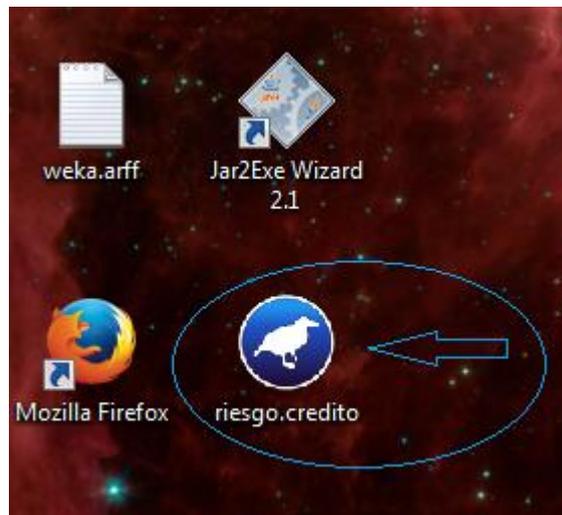


```
riesgo.credito: Bloc de notas
Archivo Edición Formato Ver Ayuda
@relation salud

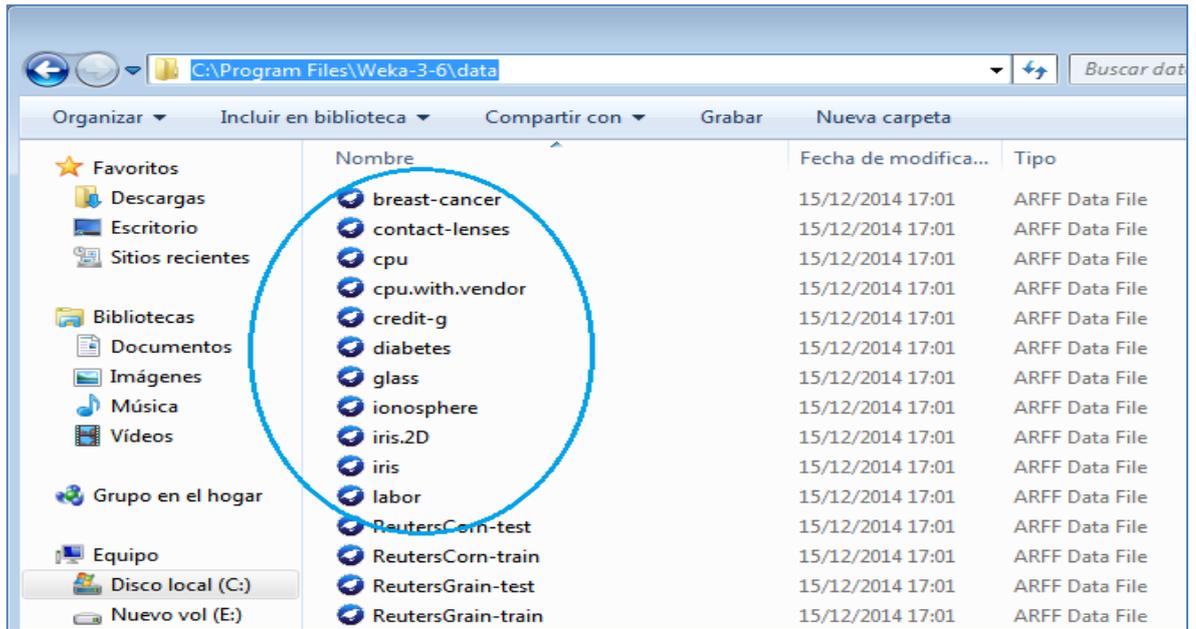
@attribute nombre STRING
@attribute estado {sano, enfermo, grave}
@attribute edad INTEGER
@attribute sexo {masculino, femenino}

@data
Alberto,sano,43,masculino
Susana,enfermo,50,femenino
Roberto,enfermo,60,masculino
Eduardo,sano,22,masculino
```

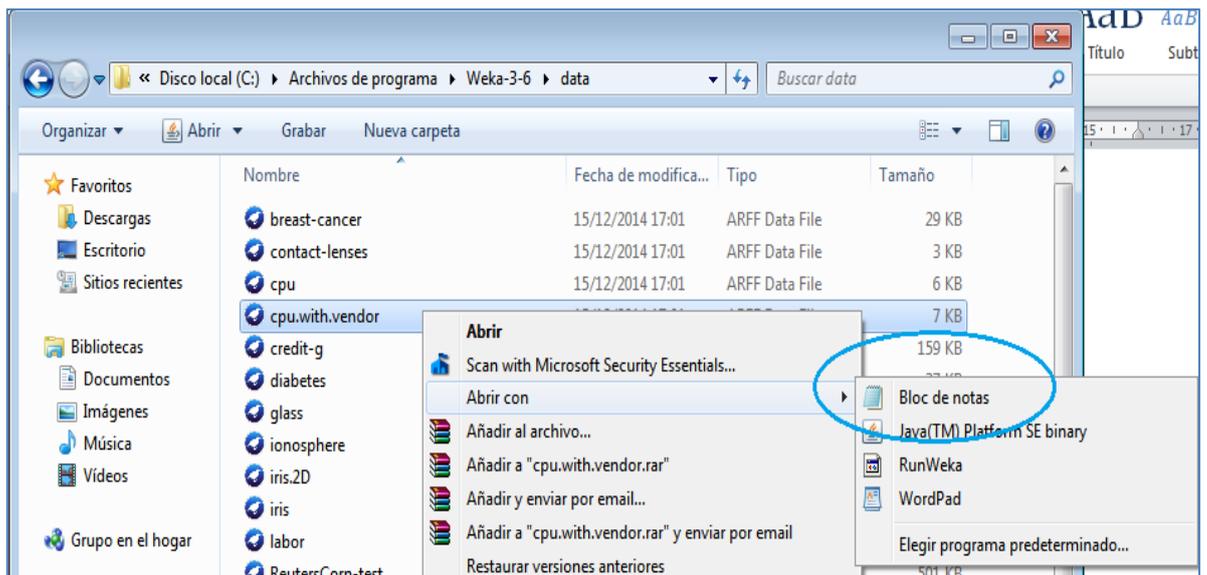
Entonces nuestro archivo debe aparecer con el formato de weka, mostrando la imagen característica, y con lo cual ya lo podemos procesar dentro de la plataforma:



2. Otra forma de crear un archivo *.arff* es tomando uno ya creado dentro de la carpeta data de weka, los cuales vienen como ejemplo para utilizarlos dentro del programa:



A cualquier archivo que hayamos elegido lo abrimos con el block de notas, dentro del mismo reemplazamos su contenido y lo guardamos con el nombre de nuestro archivo *riesgo.credito.arff*:



ANEXO 4

Archivo *riesgo.credito.arff*

@relation riesgo.credito

@attribute sexo {M, F}

@attribute edad INTEGER

@attribute ingresos REAL

@attribute dependientes INTEGER

@attribute estado.civil {casado, soltero, viudo, union_libre, divorciado}

@attribute instruccion{masterado, superior, tecnico, estudiante, secundaria, primaria, ninguna}

@attribute actividad {primaria, secundaria}

@attribute vivienda {propia, compartida, arrendada, hipotecada}

@attribute monto REAL

@attribute beneficios {IESS, seguro_de_vida, fondo_mortuario}

@attribute telefono {celular, fax, normal}

@attribute vehiculo {si, no}

@attribute tipo.cliente {socio, normal, externo}

@attribute experiencia.financiera {si, no}

@attribute mora {si, no}

@data

M,41,1100,3,casado,secundaria,primaria,hipotecada,3000,IESS,celular,si,socio,si,no
F,48,750,3,casado,secundaria,secundaria,propia,2000,IESS,celular,si,normal,si,no
F,44,600,2,casado,secundaria,secundaria,propia,1500,IESS,celular,si,normal,si,no
F,27,500,1,casado,estudiante,secundaria,arrendada,1000,IESS,celular,no,normal,no,no
F,48,1200,3,casado,superior,secundaria,hipotecada,1500,IESS,celular,si,normal,si,no
M,59,1150,3,casado,secundaria,primaria,propia,1500,IESS,celular,si,normal,si,no
F,25,354,0,soltero,estudiante,secundaria,arrendada,600,IESS,celular,no,normal,no,si
M,42,650,3,casado,superior,primaria,arrendada,1000,IESS,celular,si,normal,si,no
M,44,600,2,casado,secundaria,primaria,compartida,300,IESS,celular,si,normal,si,no
F,50,700,4,casado,secundaria,secundaria,propia,300,IESS,celular,si,normal,si,no
M,55,900,3,casado,primaria,secundaria,propia,1500,IESS,celular,no,normal,si,no
F,51,550,3,casado,secundaria,primaria,propia,1500,IESS,celular,si,normal,si,no
F,42,500,4,casado,secundaria,secundaria,arrendada,500,IESS,celular,no,normal,si,si

F,65,1200,4,divorciado,secundaria,secundaria,propia,1000,IESS,celular,si,normal,si,no
F,28,500,0,soltero,estudiante,primaria,arrendada,1545,IESS,celular,no,externo,no,si
M,30,450,2,casado,secundaria,secundaria,arrendada,1500,IESS,celular,no,normal,no,si
M,31,600,2,casado,estudiante,secundaria,compartida,300,IESS,celular,no,normal,no,si
F,50,500,6,casado,secundaria,secundaria,hipotecada,2000,IESS,celular,si,normal,si,si
M,28,550,2,casado,estudiante,secundaria,compartida,300,IESS,celular,si,normal,no,no
F,23,354,0,soltero,estudiante,secundaria,compartida,478,IESS,celular,no,normal,no,si
M,26,550,2,union_libre,tecnico,secundaria,arrendada,1100,IESS,celular,no,normal,no,no
M,23,354,0,soltero,estudiante,secundaria,arrendada,700,IESS,celular,no,normal,no,si
F,44,600,4,casado,secundaria,secundaria,hipotecada,4548,IESS,normal,no,externo,si,si
M,49,950,2,casado,secundaria,primaria,hipotecada,1000,IESS,celular,si,normal,si,no
F,45,730,3,casado,secundaria,primaria,arrendada,2019,IESS,celular,si,normal,si,no
M,46,600,5,casado,secundaria,secundaria,hipotecada,1000,IESS,celular,no,normal,si,si
F,51,1100,3,casado,secundaria,secundaria,propia,300,IESS,celular,si,normal,si,no
F,41,500,4,casado,secundaria,secundaria,arrendada,500,IESS,celular,no,normal,no,si
F,69,860,5,casado,primaria,primaria,propia,1000,IESS,celular,si,normal,si,no
M,55,600,5,casado,secundaria,primaria,propia,3015,IESS,celular,si,normal,si,si
M,77,650,7,viudo,secundaria,primaria,propia,1500,IESS,normal,no,normal,no,si
M,65,700,4,casado,primaria,secundaria,propia,1000,IESS,celular,si,normal,si,no
M,38,1130,3,casado,secundaria,secundaria,arrendada,5000,IESS,celular,si,normal,si,no
M,48,930,3,casado,secundaria,secundaria,propia,1000,IESS,celular,si,normal,si,no
F,20,650,1,casado,estudiante,secundaria,arrendada,1000,IESS,celular,no,normal,no,no
F,62,700,5,casado,primaria,primaria,hipotecada,1200,IESS,celular,si,normal,si,si
M,28,450,2,union_libre,estudiante,primaria,arrendada,1200,IESS,celular,no,normal,no,si
F,22,500,1,casado,estudiante,primaria,arrendada,1500,IESS,celular,no,normal,si,no
F,62,750,4,casado,secundaria,secundaria,propia,1000,IESS,celular,si,normal,si,si
M,64,900,5,casado,secundaria,primaria,propia,300,IESS,celular,si,normal,si,no
F,40,560,3,casado,secundaria,secundaria,arrendada,600,IESS,celular,no,normal,no,si
F,45,940,4,casado,superior,primaria,arrendada,300,IESS,celular,si,normal,si,no
F,25,354,0,soltero,estudiante,secundaria,arrendada,1500,IESS,celular,no,normal,no,no
F,26,680,2,union_libre,superior,primaria,arrendada,1000,IESS,celular,no,normal,no,no
F,66,620,6,divorciado,primaria,primaria,propia,300,IESS,celular,si,normal,si,si
F,29,354,2,casado,tecnico,secundaria,arrendada,300,IESS,celular,no,normal,no,si
M,33,354,3,casado,secundaria,secundaria,arrendada,300,IESS,celular,no,normal,no,si
M,51,1300,4,divorciado,secundaria,secundaria,propia,800,IESS,celular,si,normal,si,no

ANEXO 5

Funcionamiento de Weka

Weka es una herramienta gratuita escrita en java, la cual se la puede bajar de la siguiente dirección de acuerdo a nuestro sistema operativo, en nuestro caso utilizamos el Windows 7 profesional:

<http://www.cs.waikato.ac.nz/ml/weka/downloading.html>

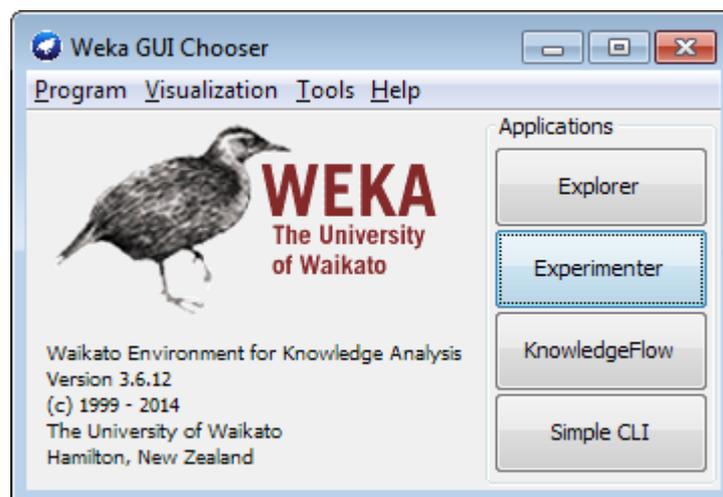
Un tutorial de utilización de weka muy interesante en español lo podemos encontrar dentro del siguiente link:

<http://ocw.uc3m.es/ingenieria-informatica/herramientas-de-la-inteligencia-artificial/contenidos/transparencias/TutorialWeka.pdf>

Una vez instalado el programa, lo ejecutamos y vamos a realizar los siguientes pasos, y por consiguiente vamos a ir ingresando a cada uno de sus componentes o subprogramas:

1. Ingreso al Programa.

La primera pantalla de weka muestra una serie de opciones dentro de Applications en la parte derecha en donde se encuentran las distintas subherramientas de weka; la que más nos interesa es Explorer (para explorar los datos)



2. El Explorer: preprocesamiento de datos (preprocess)

El Explorer permite visualizar y aplicar distintos algoritmos de aprendizaje a un conjunto de datos. Cada una de las tareas de minería de datos viene representada por una pestaña en la parte superior:

- **Preprocess:** preprocesamiento y visualización de los datos
- **Classify:** en esta sección se aplican los algoritmos de clasificación.
- **Cluster:** esta opción nos sirve para la agrupación de los datos
- **Associate:** permite la asociación de los datos
- **Select Attributes:** aquí se aplica la selección de los atributos
- **Visualize:** comprende la visualización de los datos e acuerdo a las parejas de atributos.

3. Funcionamiento de Preprocess.

La función Preprocess tiene que ver con el preprocesamiento de la información, en el cual podemos verificar y escoger los distintos algoritmos de aprendizaje para poder aplicarlos en el conjunto de datos.

En la opción Open File cargamos nuestro archivo y nos va a aparecer la siguiente pantalla:

The screenshot shows the Weka Explorer window with the 'Preprocess' tab selected. The 'Current relation' is 'riesgo.credito' with 48 instances and 15 attributes. The 'Attributes' list on the left includes 'aprobado' selected. The 'Selected attribute' panel on the right shows a bar chart for 'aprobado' with two categories: 'si' (27) and 'no' (21).

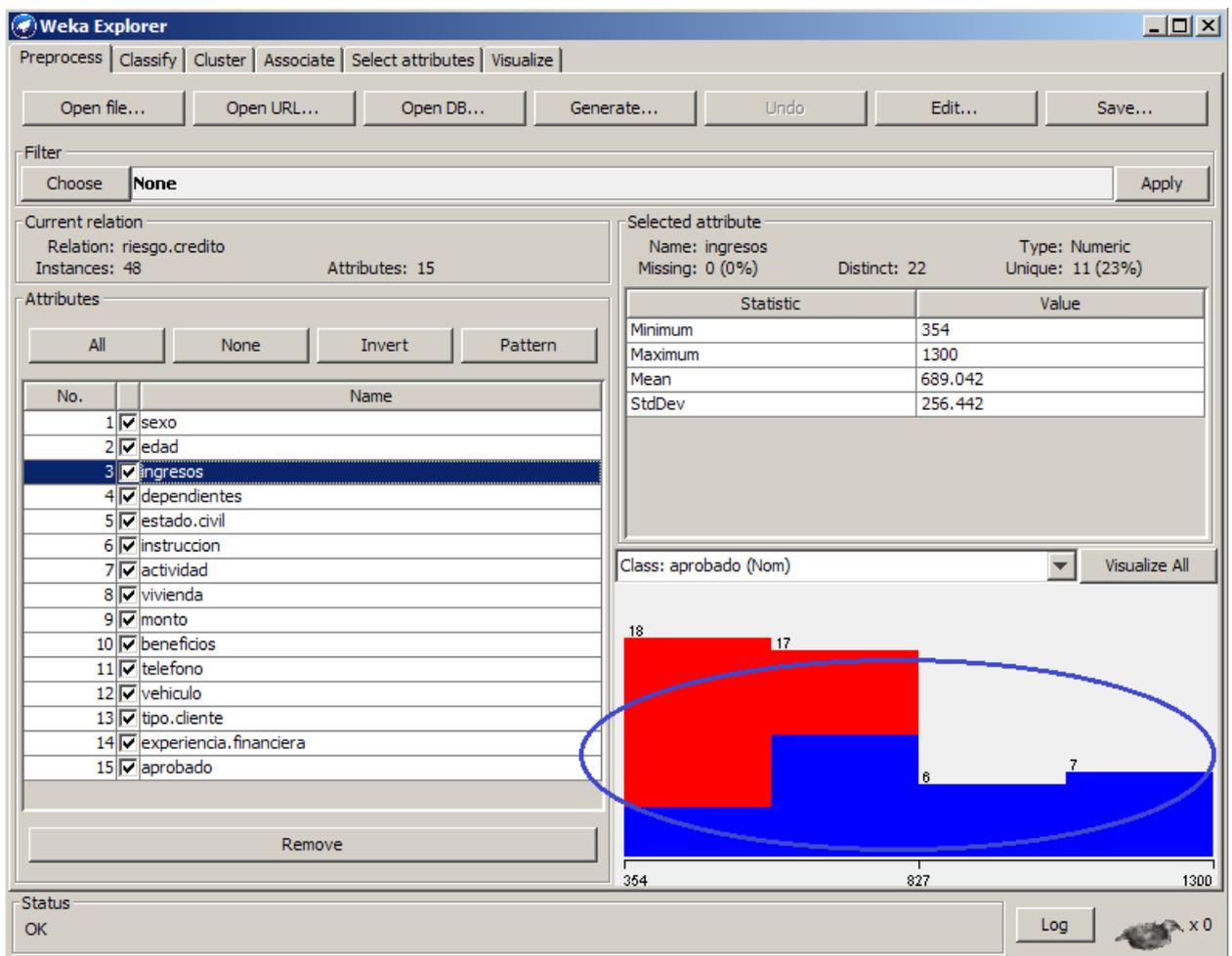
No.	Label	Count
1	si	27
2	no	21

En la parte superior derecha aparece información estadística sobre los atributos (media, desviación típica, valores máximo y mínimo, Unique se refiere al número de valores que sólo

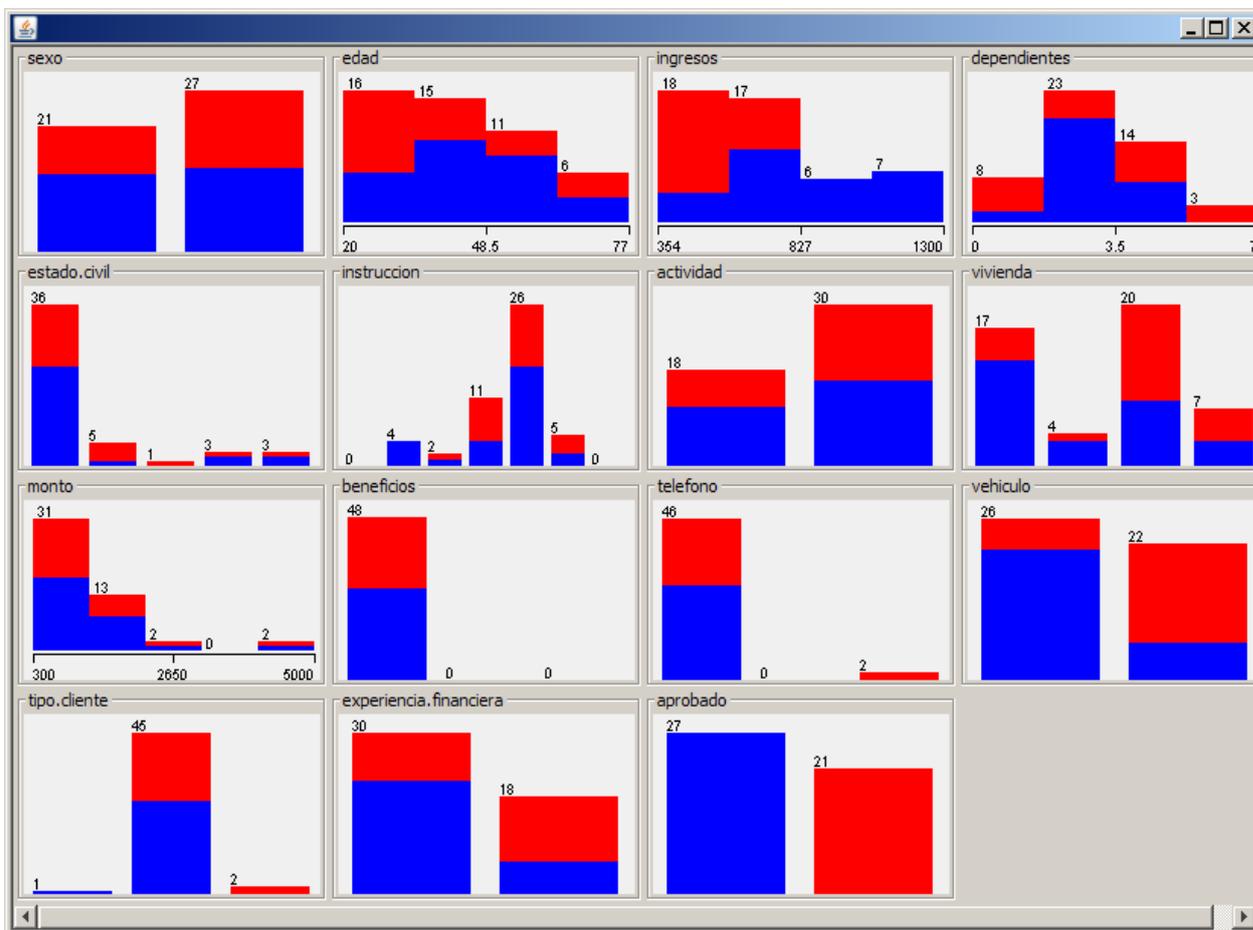
aparecen una vez en ese atributo y distinct al número de valores distintos (o sea, excluyendo los valores repetidos). Un poco más abajo, aparece el desglose de los valores del atributo por clase. En el eje X aparecen los valores del atributo seleccionado en la parte de la izquierda Recordemos que hay tres posibles clases, que en el gráfico aparecen representadas con los colores azul oscuro, rojo y azul claro.

En el gráfico de la parte inferior derecha aparecen los valores de los atributos divididos por su clase, en el eje X nos detalla los valores del atributo y en el eje Y la participación de las instancias en el rango de los valores de los atributos. Por ejemplo, en el grafico siguiente, tomamos como ejemplo el atributo ingresos, en donde:

- 18 personas perciben un sueldo de 354 \$ hasta 590.5 \$.
- 17 personas tienen un sueldo de 590.5 \$ hasta 827 \$.
- 6 personas tienen un sueldo desde 827 \$ hasta 1063.5 \$.
- 7 personas perciben un ingreso desde 1063.5 \$ hasta 1300 \$



Si queremos revisar los detalles de los atributos que se utilizan en la gestión de créditos de los clientes en la cooperativa, pulsamos la opción Visualize All:



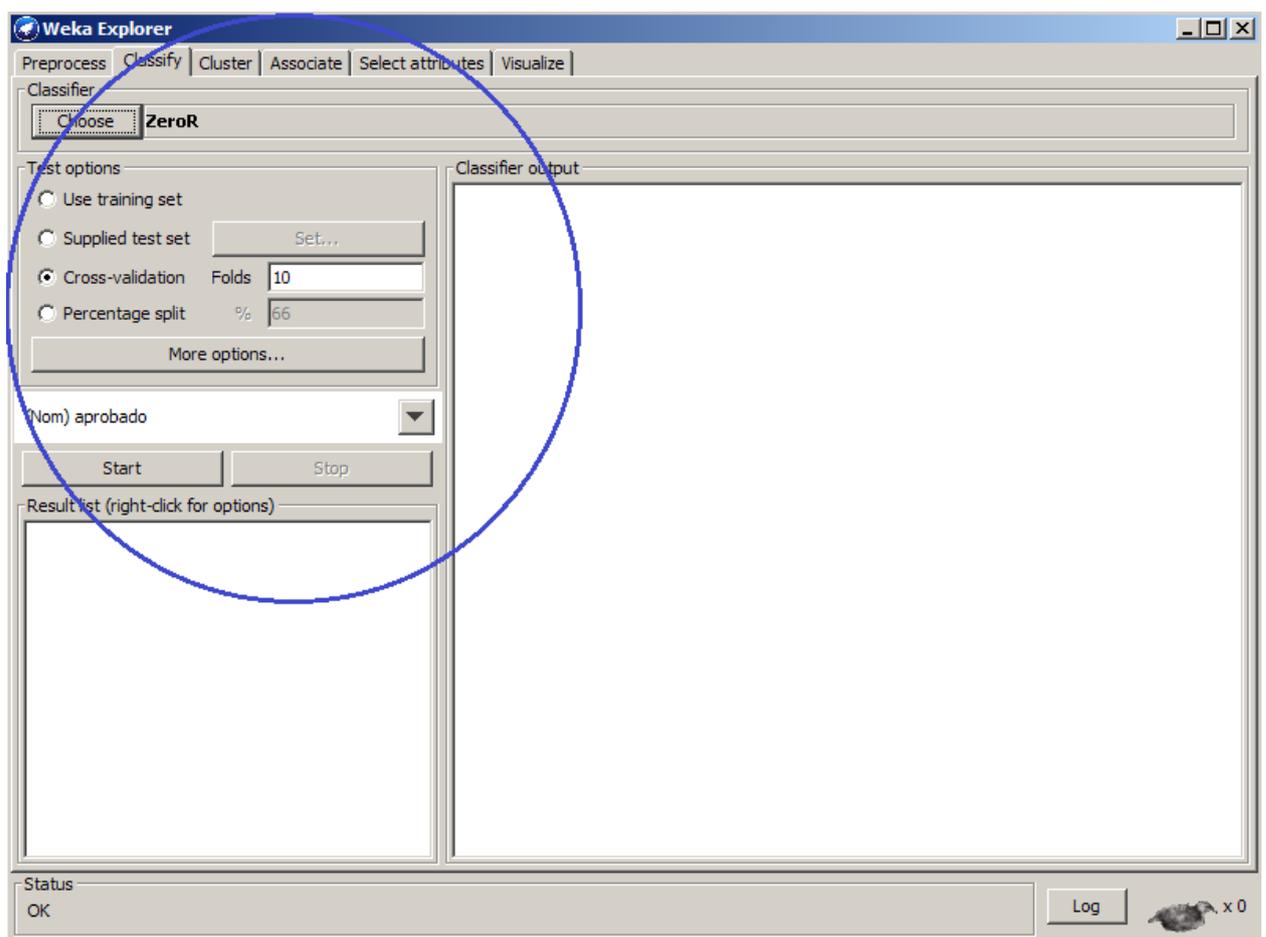
Esta visualización puede ser útil para comprobar cómo de efectivo es cada uno de los atributos, considerados por separado. Recordemos que el objetivo de este dominio es encontrar un buen clasificador, es decir, una frontera que separe los datos pertenecientes a distintas clases; por ejemplo en este caso vemos que el atributo *aprobado* separe perfectamente los 27 casos de los 21 restantes.

4. Funcionamiento de Classify.

La función Classify dentro de Weka nos sirve para realizar la clasificación de los datos que poseemos; para ello tenemos que elegir un clasificador mediante el botón Choose, ya que por defecto nos aparece ZeroR. Debajo tenemos las opciones de test options, en donde vamos a procesar el porcentaje esperado de aciertos, entre estas opciones tenemos:

- **Use training set:** realiza un procesamiento esencial, no es muy recomendable usarlo.

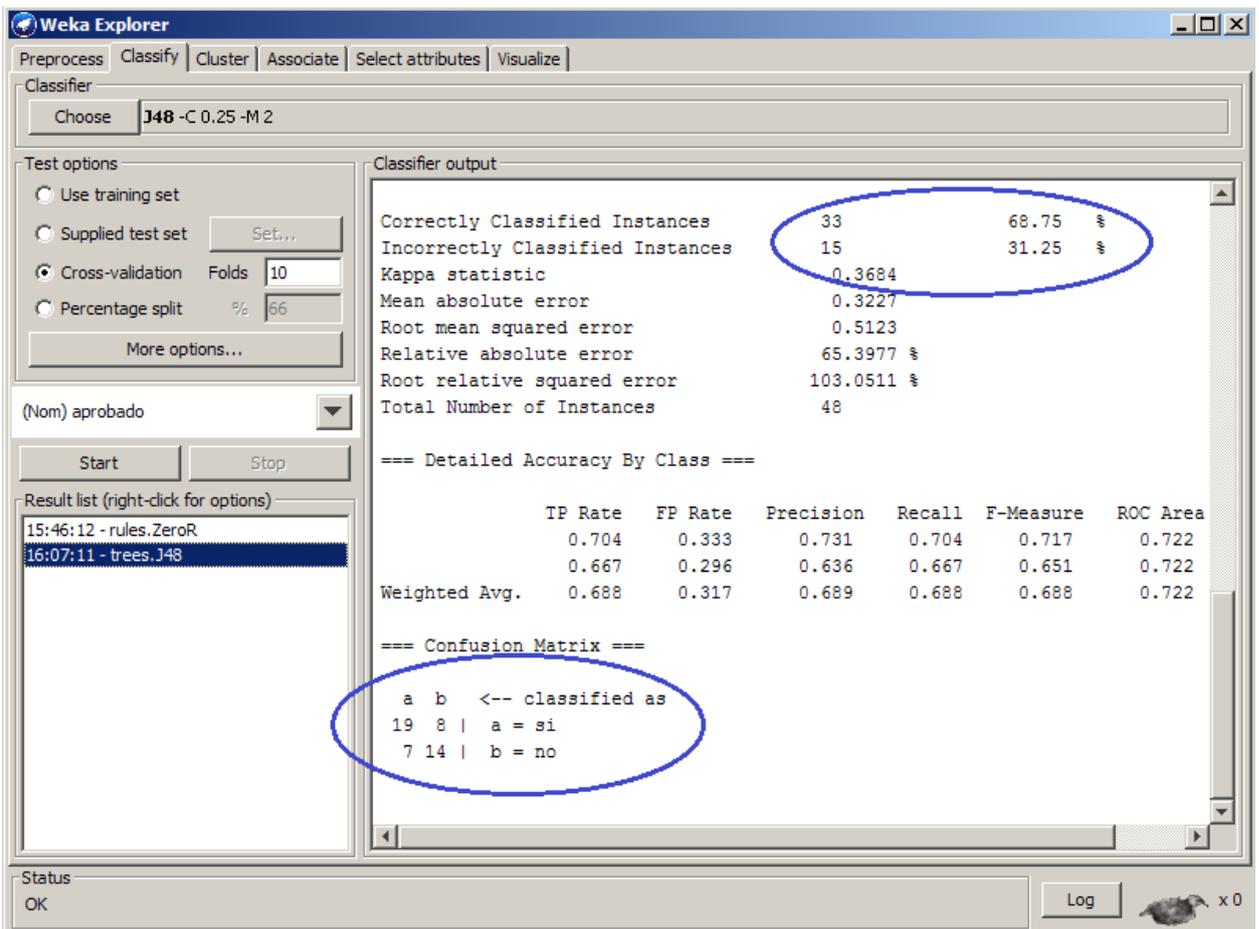
- **Supliedd test set:** para usar un archivo de datos diferente al que estamos procesando.
- **Cross-validation:** nos sirve para realizar el cálculo de aciertos esperado, usando una validación cruzada estratificada de un número de participaciones dadas (Folder), de cual la cantidad por omisión es 10.
- **Percentage Split:** divide el conjunto de entrenamiento en dos partes, una para realizar el clasificador y la otra para realizar el test. En este caso, se dividirá el conjunto de entrenamiento que ya habíamos seleccionado en la pestaña de Preprocess (el riesgo.credito.arff) y se dividirá en dos partes: los primeros 66% de los datos para construir el clasificador y el 33% finales, para hacer el test. Podemos seleccionar el porcentaje para entrenamiento (por omisión, es de 66%). Importante: al utilizar esta opción, Weka desordena aleatoriamente el conjunto inicial(en este caso, el riesgo.credito.arff) y después parte en 66% para entrenamiento y 33% para test. De esta manera, si construyéramos el clasificador dos veces, obtendríamos dos desordenaciones distintas, y por tanto dos porcentajes de aciertos en test ligeramente distintos



Dentro de Weka tenemos una amplia variedad de Clasificadores, podemos escoger el que nos presente más aciertos en la clasificación de nuestros datos. En este momento, está seleccionado ZeroR (arriba a la izquierda). Este clasificador clasifica a todos los datos con la clase de la clase mayoritaria. Es decir, si el 90% de los datos son positivos y el 10% son negativos, clasificará a todos los datos como positivos. Es conveniente utilizar primero este clasificador, porque el porcentaje de aciertos que obtengamos con el, es el que habrá que superar con el resto de clasificadores. Antes de lanzarlo, seleccionemos la opción Crossvalidation (10) para hacer el test.

5. Algoritmo J48.

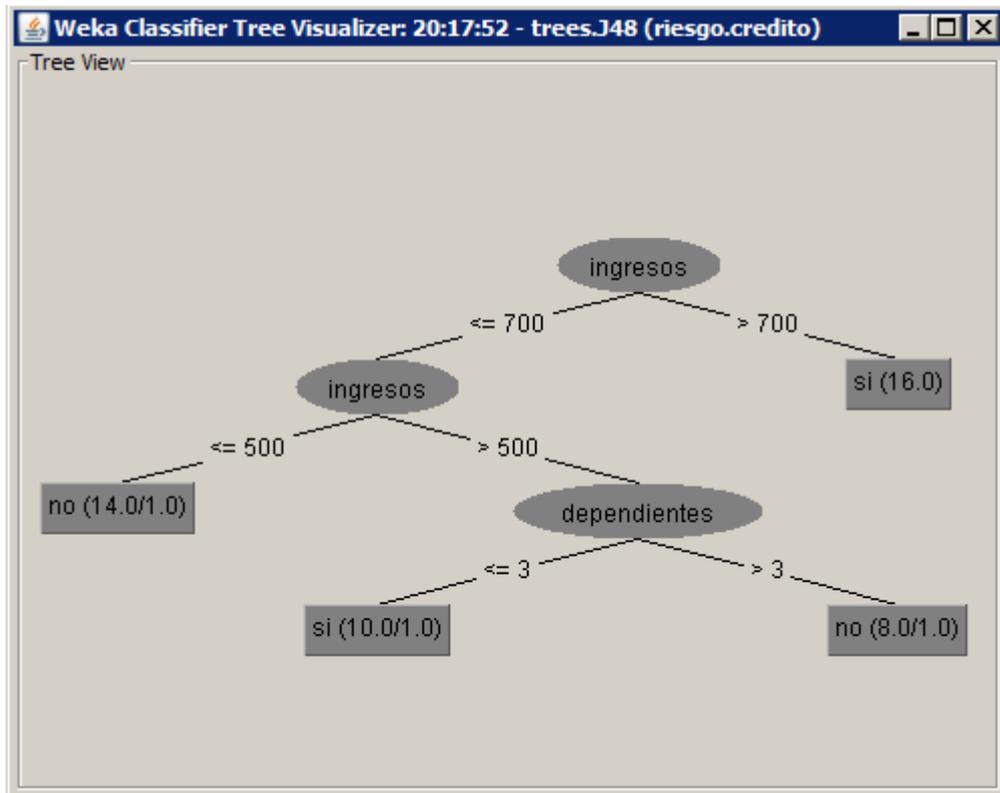
Este algoritmo es uno de los más utilizados en Weka para la minería de datos basado en arboles de decisión; se trata de un clasificador sencillo, rápido y ha dado buenos resultados en comparación con otros algoritmos más complejos; una de sus características principales es que selecciona el atributo o atributos que mejor explican la clase de salida.



El algoritmo J48 de WEKA es una implementación del algoritmo C4.5, uno de los algoritmos de minería de datos más utilizado. Se trata de un refinamiento del modelo generado con

OneR. Supone una mejora moderada en las prestaciones, y se podrá conseguir una probabilidad de acierto bastante elevada en pequeñas y grandes cantidades de información.

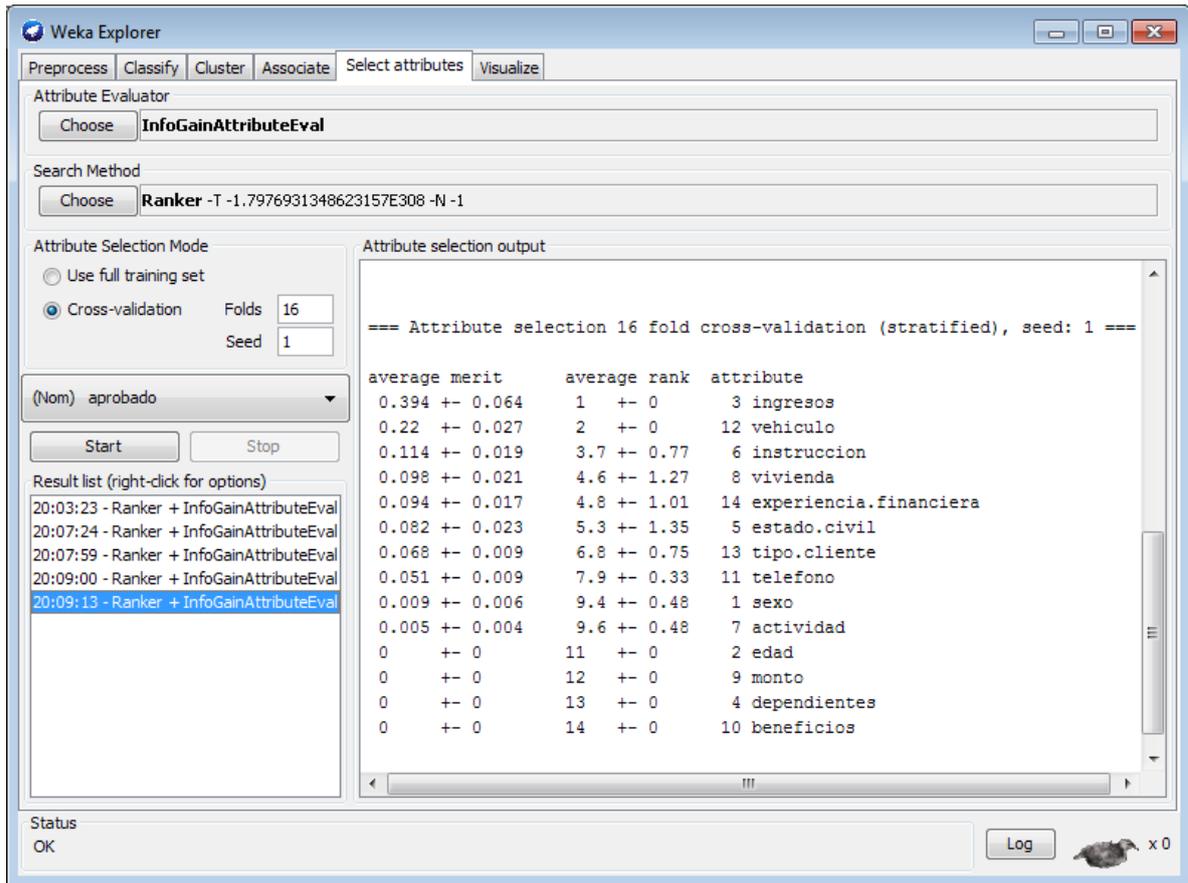
Con el algoritmo J48 también se puede visualizar el árbol decisión dentro de weka:



6. Selección de atributos relevantes.

Mediante la selección de atributos en Weka vamos a eliminar los que nos resulten redundantes e irrelevantes; ya que al existir un número elevado de atributos puede ser que dentro del modelo se produzca complejidad y no nos conduzca a la obtención del resultado esperado. Escogiendo los atributos adecuados podemos hacer que el modelo sea más efectivo y sencillo. Dentro de Weka podemos escoger las tres posibilidades:

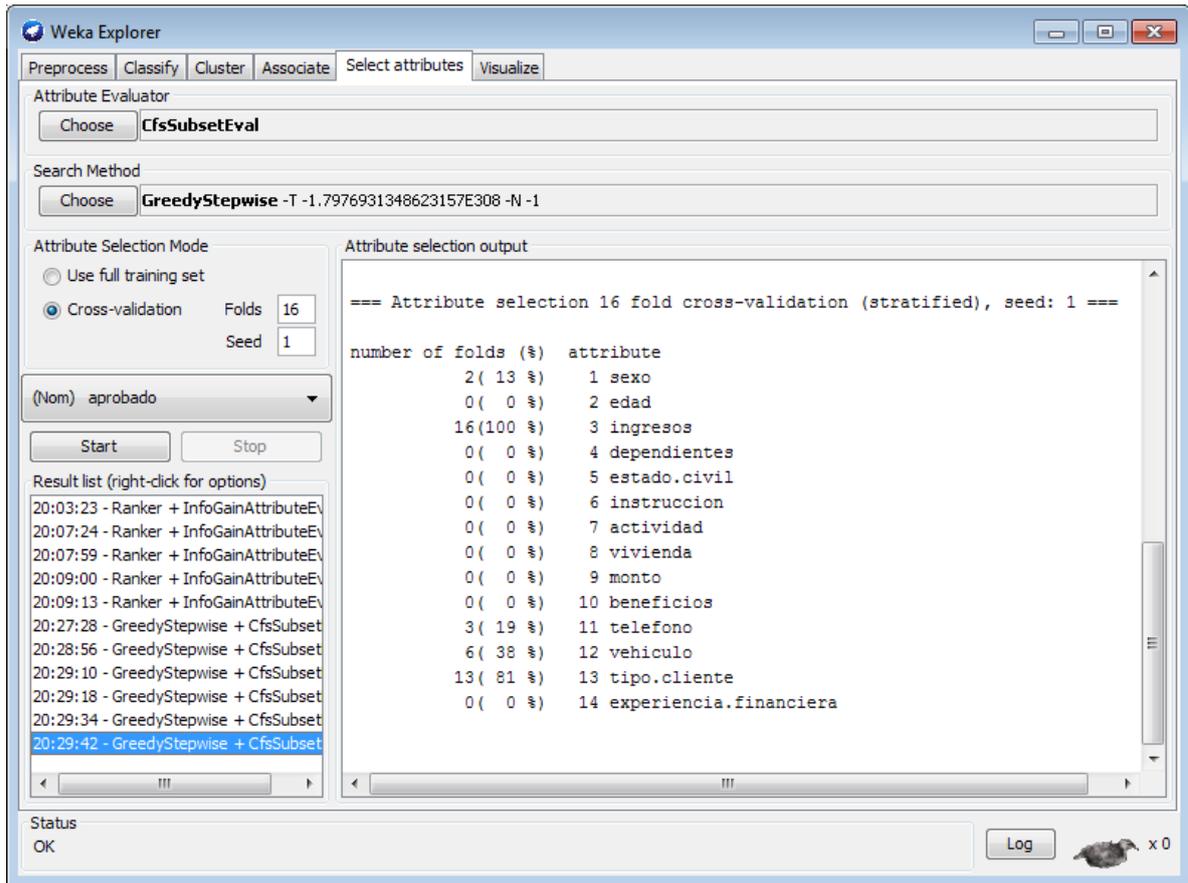
- a. Evaluación de atributos: con el Método de búsqueda = Ranker y Método de Evaluación = InfoGainAttributeEval.



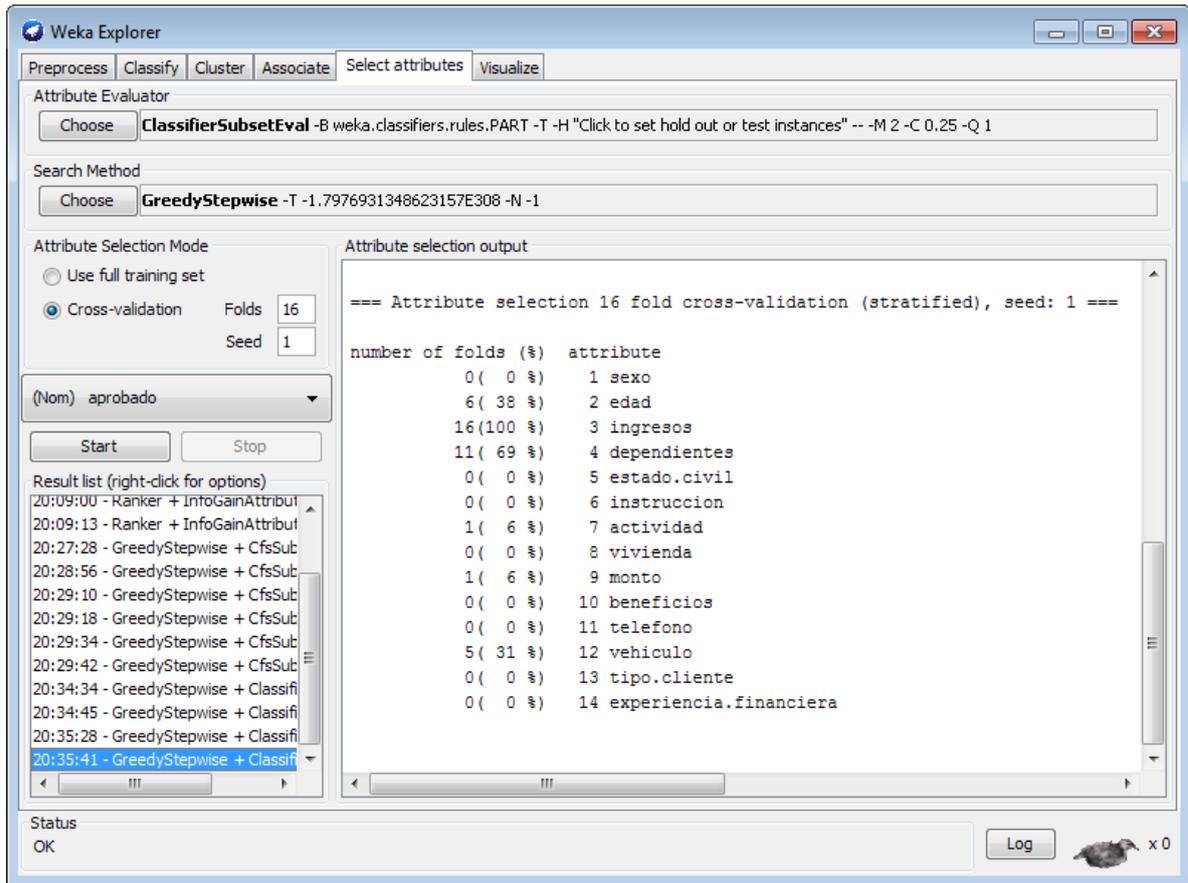
Ranker nos ordena los atributos y desglosa los valores de cada atributo por clase. Weka nos da dos informaciones: el average merit (y su desviación típica) y el average rank (y su desviación típica). El primero se refiere a la media de las correlaciones (medidas con InfoGain) en los ciclos de validación cruzada

- b. Metodo Filter: con el Método de búsqueda = Greedy Stepwise y Método de evaluación = CfsSubsetEval.

Recordemos que CfsSubsetEval selecciona subconjuntos de atributos.



- c. Método Wrapper: Método de búsqueda = Greedy Stepwise y Método de evaluación = ClassifierSubsetEval.

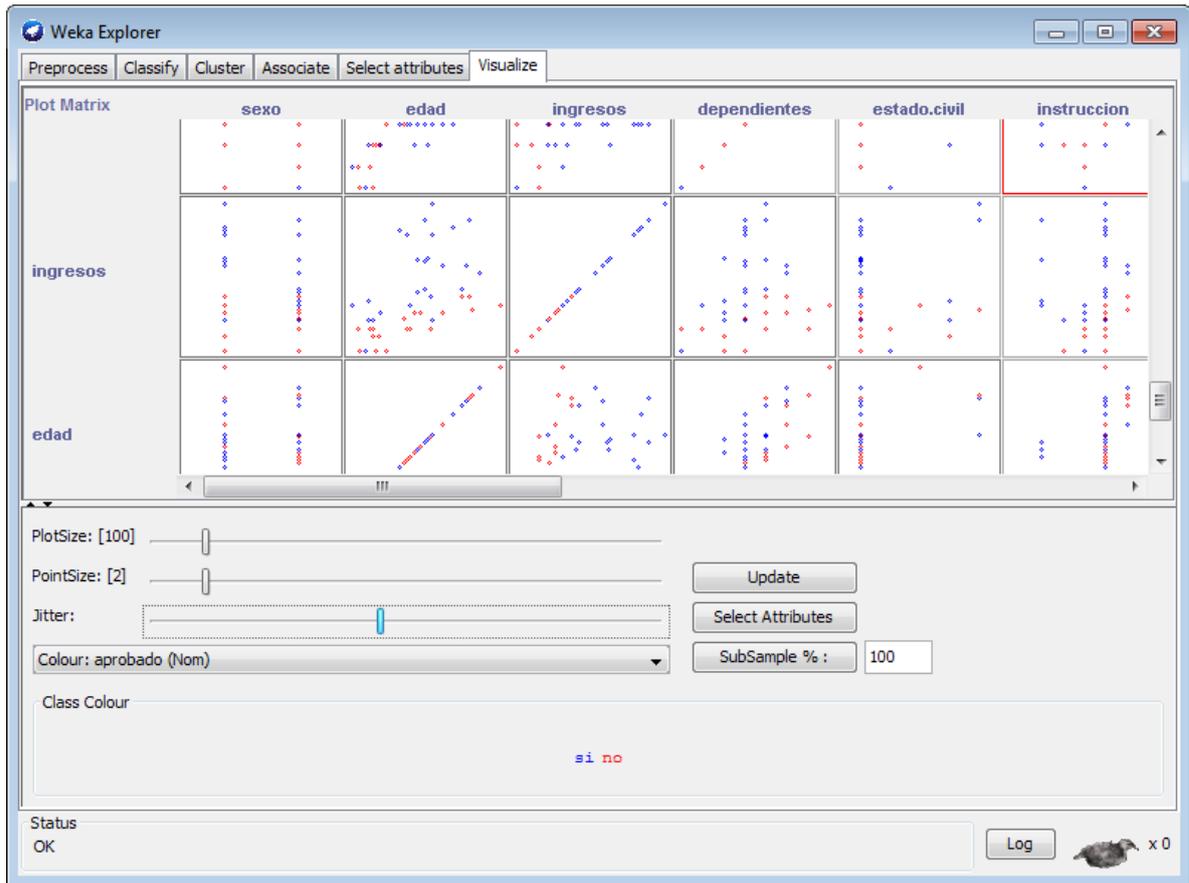


Mucho cuidado, porque un método Wrapper, requiere que se seleccione un clasificador base. Para ello, pincharemos sobre ClassifierSubsetEval para ver sus parámetros, y veremos que hay uno que es Classifier y que por omisión tiene el valor ZeroR. Pongamos por ejemplo el clasificador PART

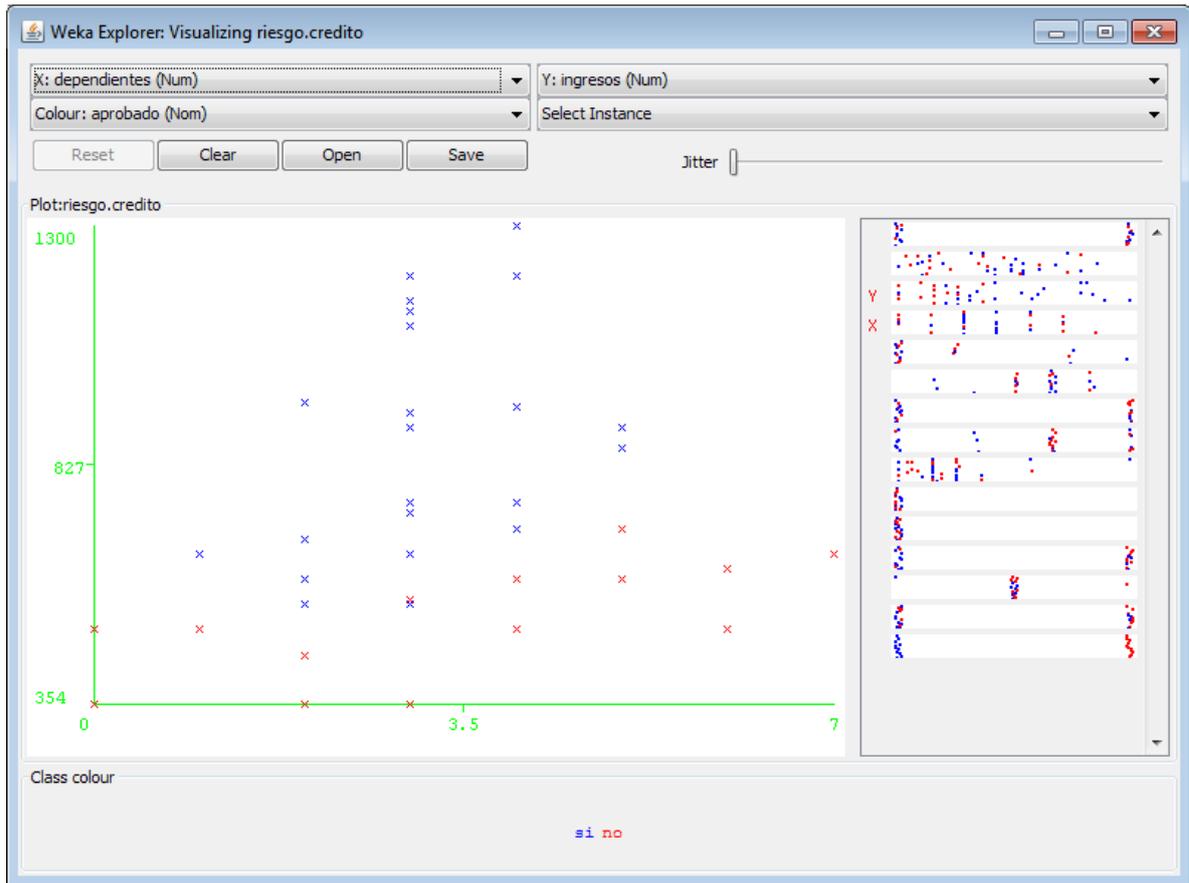
Seguidamente, dentro de la herramienta vamos a borrar los atributos redundantes y solamente vamos a trabajar con los escogidos para realizar la comprobación final.

7. Visualize.

Un aspecto importante dentro de la herramienta de Weka es la visualización de resultados; para ello nos vamos a la pestaña de Visualize, en donde vamos a poder determinar la relación de los atributos por parejas:



En esta ventana vamos a verificar un par atributos relevantes, por ejemplo: *ingresos* y *dependientes*, en donde podemos ver que se encuentran casi perfectamente separados por lo que se deduce que son factibles de clasificación:



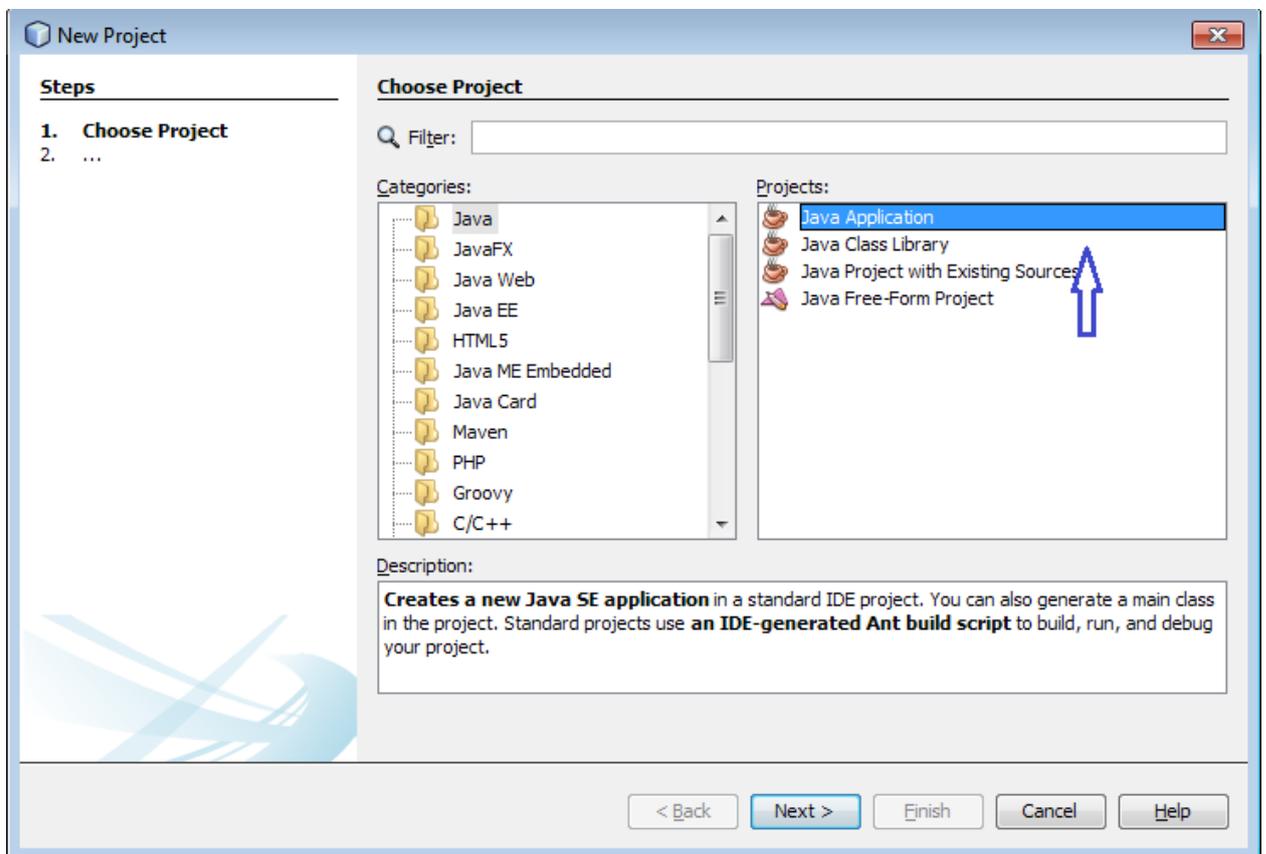
Hasta ahora, hemos podido ver cuáles son los atributos más relevantes, y hemos podido visualizar que tal separan las clases, por parejas. Pero todavía no hemos observado si la selección de atributos mejora (o empeora) el porcentaje de aciertos esperado, o si los modelos que se construyen son más complejos o más simples. Lo que necesitamos es un meta-clasificador que primero pase un filtro de selección de atributos, y después realice aprendizaje (y test) utilizando exclusivamente los atributos seleccionados. Dicho metaclasificador se llama attribute selected classifier y lo encontraremos entre los classifiers meta, volviendo a la pestaña classify (donde usamos antes el PART y el J48).

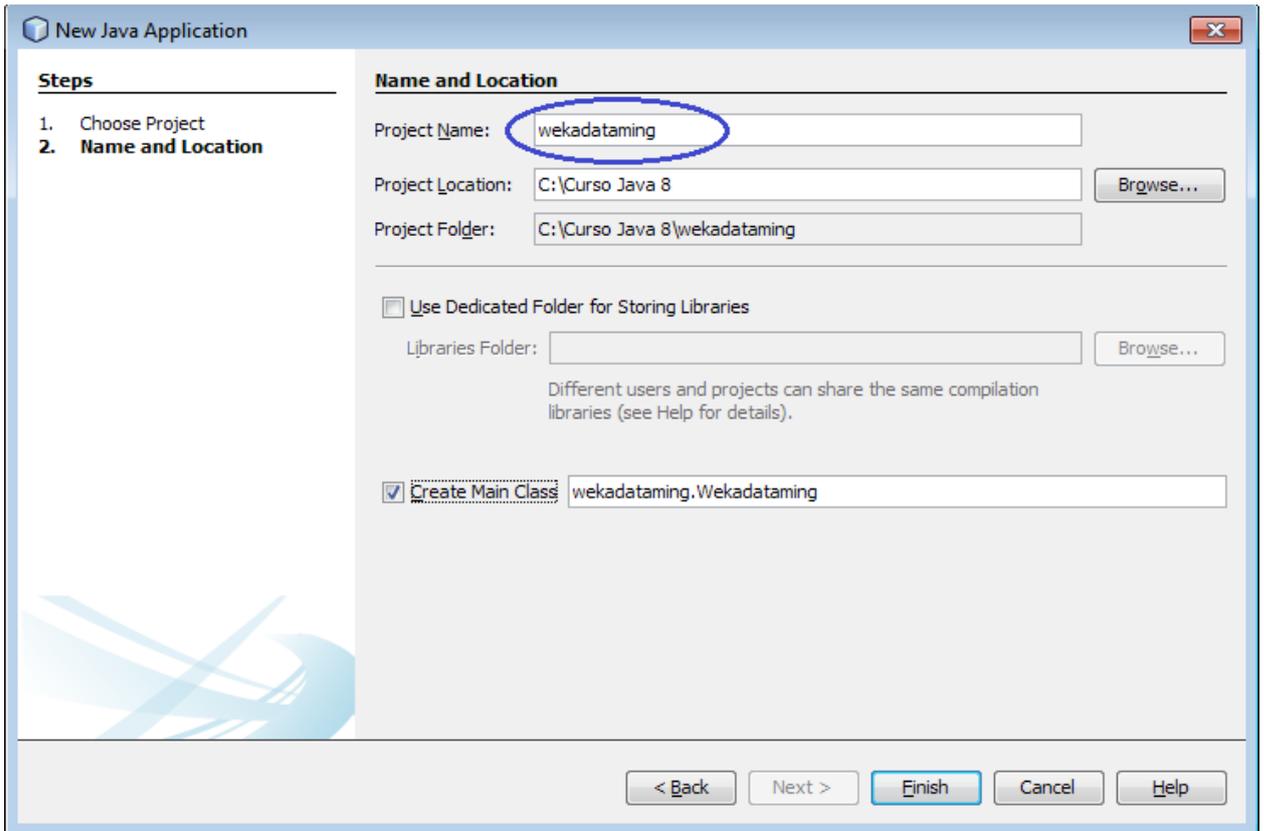
ANEXO 6

Construcción del proyecto wekadataminig en Java Netbeans.

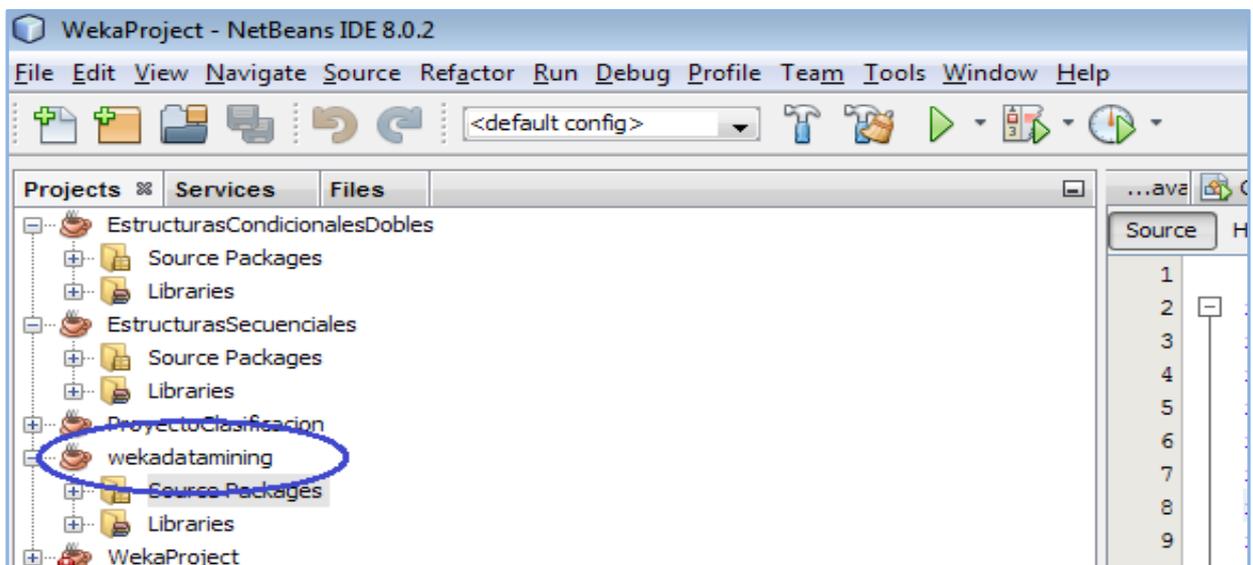
Weka se encuentra desarrollado completamente en Java y vamos a poder exportar y utilizar sus API y librerías dentro de nuestro aplicativo. La versión de Java Netbeans que vamos a utilizar es la IDE 8.0.2, la última que se encontró disponible y se la puede bajar de la siguiente dirección: <https://netbeans.org/downloads/>

Para poder implementar nuestro modelo, se ha creará inicialmente el nombre de nuestro proyecto: wekadataminig; escogiendo previamente la categoría Java y dentro de ésta la opción Java Application, según las pantallas siguientes:





Una vez creado el nombre del proyecto, éste aparece dentro de la pantalla principal de Java en donde podemos definir nuestras clases, librerías y demás programas.



Una vez creado nuestro proyecto vamos a necesitar dos componentes esenciales dentro del mismo, uno que nos permita manipular los algoritmos de clasificación de Weka y otro que nos sirva para presentar los resultados finales al usuario con un entorno amigable.

ANEXO 7

Codigo fuente del JFrame: *Predecir*

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package modulos;

import java.io.FileOutputStream;
import java.io.IOException;
import java.sql.*;
import javax.swing.JOptionPane;
import java.util.Vector;
import java.util.*;
import org.apache.poi.ss.usermodel.*;
import org.apache.poi.ss.util.CellRangeAddress;
import org.apache.poi.hssf.usermodel.HSSFWorkbook;

/**
 *
 * @author romel gaona
 */
public class Predecir extends javax.swing.JFrame {

    /**
     * Creates new form Predecir
     */
    public Predecir() {
        initComponents();
    }

    /**
     * This method is called from within the constructor to initialize the form.
     * WARNING: Do NOT modify this code. The content of this method is always

```

```

* regenerated by the Form Editor.
*/
@SuppressWarnings("unchecked")
// <editor-fold defaultstate="collapsed" desc="Generated Code">
private void cbeneficiosActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
}
private void ctelefonoActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
}
private void cvehiculoActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
}
private void ctipoclienteActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
}
private void cexpfinancieraActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
}
private void caprobadoActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
}
private void csexoActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
}
private void bttnpredecirActionPerformed(java.awt.event.ActionEvent evt) {
    // delaracion de variables:
    Integer edad;
    Double ingresos;
    Integer dependientes;
    String vehiculo;
    Double monto;
    String aprobado;
    aprobado = "\n";
    String actual;
    //entrada de datos

```

```

ingresos = Double.parseDouble(cingresos.getText());
monto = Double.parseDouble(cmonto.getText());
dependientes = Integer.parseInt(cdependientes.getText());
edad = Integer.parseInt(cedad.getText());
vehiculo = cvehiculo.getText();
actual = caprobado.getText();
//aprobado = caprobado.getText();
//proceso
if (ingresos <= 750){
    if (dependientes <= 4){
        if (vehiculo == "si"){
            aprobado = "si";
        }
        else{
            if (edad <= 27){
                if(monto <= 800){
                    aprobado = "no";
                }
                else{
                    aprobado = "si";
                }
            }
            else{
                aprobado = "no";
            }
        }
    }
    else {
        aprobado = "no";
    }
}
else{
    aprobado = "si";
}
//resultado.setText("ACTUAL DESICION:" +actual);

```

```

        aresultado.setText("RESOLUCION: " + actual + " SUGERENCIA: " +
        aprobado);
        //resultado.setText("SUGERENCIA:" + aprobado);
    }

```

```

private void btngrabarActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:

```

```

        Workbook libro = new HSSFWorkbook();
        Sheet hoja = libro.createSheet("Hoja1");
        Row fila = hoja.createRow(0);
        Cell celda = fila.createCell(0);
        celda.setCellValue(csexo.getText());
        Cell celda1 = fila.createCell(1);
        celda1.setCellValue(cedad.getText());
        Cell celda2 = fila.createCell(2);
        celda2.setCellValue(cingresos.getText());
        Cell celda3 = fila.createCell(3);
        celda3.setCellValue(cdependientes.getText());
        Cell celda4 = fila.createCell(4);
        celda4.setCellValue(cestadocivil.getText());
        Cell celda5 = fila.createCell(5);
        celda5.setCellValue(cinstruccion.getText());
        Cell celda6 = fila.createCell(6);
        celda6.setCellValue(cactividad.getText());
        Cell celda7 = fila.createCell(7);
        celda7.setCellValue(cvivienda.getText());
        Cell celda8 = fila.createCell(8);
        celda8.setCellValue(cmonto.getText());
        Cell celda9 = fila.createCell(9);
        celda9.setCellValue(cbeneficios.getText());
        Cell celda10 = fila.createCell(10);
        celda10.setCellValue(ctelefono.getText());
        Cell celda11 = fila.createCell(11);
        celda11.setCellValue(cvehiculo.getText());
        Cell celda12 = fila.createCell(12);

```

```

        celda12.setCellValue(ctipocliente.getText());
        Cell celda13 = fila.createCell(13);
        celda13.setCellValue(cexpfinanciera.getText());
        Cell celda14 = fila.createCell(14);
        celda14.setCellValue(caprobado.getText());
        String file = "Nuevos_Clientes.xls";
        try (FileOutputStream out = new FileOutputStream(file)){
            libro.write(out);
        }
        catch(IOException e){
            System.err.println(e.getMessage());
        }
        //out.close();
    }

private void btnsalirActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    System.exit(0);
}

private void btnregresarActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    Principal principal = new Principal();
    principal.setVisible(true);
    this.dispose();
}

private void bttnlimpiarActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    csexo.setText("");
    cedad.setText("");
    cingresos.setText("");
    cdependientes.setText("");
    cestadocivil.setText("");
}

```

```

    cinstruccion.setText("");
    cactividad.setText("");
    cvivienda.setText("");
    cmonto.setText("");
    cbeneficios.setText("");
    ctelefono.setText("");
    cvehiculo.setText("");
    ctipocliente.setText("");
    cexpfinciera.setText("");
    caprobado.setText("");
}

/**
 * @param args the command line arguments
 */
public static void main(String args[]) {
    /* Set the Nimbus look and feel */
    //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">
    /* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and
    feel.
    *For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
    */
    try {
        for (javax.swing.UIManager.LookAndFeelInfo info :
            javax.swing.UIManager.getInstalledLookAndFeels()) {
            if ("Nimbus".equals(info.getName())) {
                javax.swing.UIManager.setLookAndFeel(info.getClassName());
                break;
            }
        }
    } catch (ClassNotFoundException ex) {

        java.util.logging.Logger.getLogger(Predecir.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    } catch (InstantiationException ex) {

```

```
java.util.logging.Logger.getLogger(Prededir.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
```

```
    } catch (IllegalAccessException ex) {
```

```
java.util.logging.Logger.getLogger(Prededir.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
```

```
    } catch (javax.swing.UnsupportedLookAndFeelException ex) {
```

```
java.util.logging.Logger.getLogger(Prededir.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
```

```
    }
```

```
//</editor-fold>
```

```
/* Create and display the form */
```

```
java.awt.EventQueue.invokeLater(new Runnable() {
```

```
    public void run() {
```

```
        new Prededir().setVisible(true);
```

```
    }
```

```
});
```

```
}
```

```
// Variables declaration - do not modify
```

```
private javax.swing.JTextArea are resultado;
```

```
private javax.swing.JButton bttngrabar;
```

```
private javax.swing.JButton bttnlimpiar;
```

```
private javax.swing.JButton bttnprededir;
```

```
private javax.swing.JButton bttnregresar;
```

```
private javax.swing.JButton bttnsalir;
```

```
private javax.swing.JTextField cactividad;
```

```
private javax.swing.JTextField caprobado;
```

```
private javax.swing.JTextField cbeneficios;
```

```
private javax.swing.JTextField cdependientes;
```

```
private javax.swing.JTextField cedad;
```

```
private javax.swing.JTextField cestadocivil;
```

```
private javax.swing.JTextField cexpfinanciera;
```

```
private javax.swing.JTextField cingresos;
private javax.swing.JTextField cinstruccion;
private javax.swing.JTextField cmonto;
private javax.swing.JTextField csexo;
private javax.swing.JTextField ctelefono;
private javax.swing.JTextField ctipocliente;
private javax.swing.JTextField cvehiculo;
private javax.swing.JTextField cvivienda;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel10;
private javax.swing.JLabel jLabel11;
private javax.swing.JLabel jLabel12;
private javax.swing.JLabel jLabel13;
private javax.swing.JLabel jLabel14;
private javax.swing.JLabel jLabel15;
private javax.swing.JLabel jLabel2;
private javax.swing.JLabel jLabel3;
private javax.swing.JLabel jLabel4;
private javax.swing.JLabel jLabel5;
private javax.swing.JLabel jLabel6;
private javax.swing.JLabel jLabel7;
private javax.swing.JLabel jLabel8;
private javax.swing.JLabel jLabel9;
private javax.swing.JPanel jPanel1;
private javax.swing.JPanel jPanel2;
private javax.swing.JScrollPane jScrollPane1;
// End of variables declaration
}
```