UNIVERSIDAD DE CHILE
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
DEPARTAMENTO DE INGENIERÍA MATEMÁTICA

DETECCIÓN DE FRAUDE EN TARJETAS DE CRÉDITO

MEMORIA PARA OPTAR AL TÍTULO DE
INGENIERO CIVIL MATEMÁTICO

DARÍO ESTEBAN CEPEDA GARCÍA

SANTIAGO DE CHILE
JULIO 2012

UNIVERSIDAD DE CHILE
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
DEPARTAMENTO DE INGENIERÍA MATEMÁTICA

DETECCIÓN DE FRAUDE EN TARJETAS DE CRÉDITO

MEMORIA PARA OPTAR AL TÍTULO DE
INGENIERO CIVIL MATEMÁTICO

DARÍO ESTEBAN CEPEDA GARCÍA

PROFESOR GUÍA:
ALEJANDRO JOFRÉ CÁCERES

MIEMBROS DE LA COMISIÓN:
JOAQUÍN FONTBONA TORRES
RICHARD WEBER HAAS

SANTIAGO DE CHILE
JULIO 2012

RESUMEN DE LA MEMORIA
PARA OPTAR AL TÍTULO DE
INGENIERO CIVIL MATEMÁTICO
POR: DARÍO ESTEBAN CEPEDA GARCÍA
FECHA: JULIO 2012
PROF. GUÍA: ALEJANDRO JOFRÉ CÁCERES

DETECCIÓN DE FRAUDE EN TARJETAS DE CRÉDITO

Resumen

El presente trabajo de memoria propone una metodología para la construcción de un modelo de predicción de fraude transaccional, el cual tiene como objetivo identificar aquellas transacciones que presentan mayor probabilidad de ser fraudulentas. Las transacciones son hechas con una tarjeta de crédito asociada a una empresa del área *Retail* (empresa dedicada a la venta al detalle).

El fraude en tarjetas de crédito es un problema serio, creciente, complejo y dinámico. El gran volumen de ventas de las empresas del área *Retail*, la diversificación de los comercios en los cuales participa y el rápido crecimiento de la popularidad de ventas online, hace que existan diversas formas en las que un cliente pueda verse afectado por fraude.

Los datos disponibles se encuentran en tres bases de datos de la empresa. El problema con estas bases es que las transacciones fraudulentas no se encuentran marcadas, éstas últimas se encuentran en otra base. Por falta de un identificador común en las bases, se tienen que marcar manualmente las transacciones fraudulentas.

Luego de marcar los casos fraudulentos, la solución propuesta corresponde a construir varios modelos de clasificación binaria, los que asignan a cada transacción una probabilidad de ser fraudulenta. Esta asignación se realizó sobre la base de la definición de un patrón característico mediante un conjunto de variables de entrada, siendo éstas definidas en conjunto con los expertos del negocio.

Para la construcción de los modelos se usan cuatro técnicas distintas: *support vector machines*, redes neuronales artificiales, árboles de decisión y regresión logística. Se consideran modelos aplicados en el total de las transacciones y también agrupando rubros específicos, para así medir cómo cambia la predicción al segmentar. En la construcción de los modelos se usaron distintas proporciones de transacciones normales y fraudulentas, con el objetivo de encontrar qué proporción es mejor para la detección.

Como conclusión general, modelos más complejos como *support vector machines* y redes neuronales artificiales tienen mejor rendimiento que modelos más sencillos como regresión logística. Cuando se disminuye la proporción de transacciones fraudulentas en la construcción de los modelos se obtiene una mejor predicción. Al segmentar por rubros específicos se obtienen aún mejores resultados, esto muestra que es mejor segmentar y tener varios modelos que uno solo para todas las transacciones.

Índice general

Índice de figuras

Índice de tablas

Capítulo 1

Introducción

El fraude en tarjetas crédito es un problema serio y que crece día a día. Con el aumento de éstas como medio de pago, debido al rápido incremento de las ventas online y el cambio de comportamiento de las personas al pagar, se ha producido una mayor exposición al fraude transaccional.

Además, con el paso de los años y la evolución de los métodos para detectar fraude, las personas que lo cometen también han evolucionado sus prácticas para evitar la detección, por lo que el fraude es dinámico, siempre está cambiando. En consecuencia, los métodos de detección de fraude necesitan ser constantemente mejorados.

Dentro del comportamiento del fraude, se quiere encontrar invariantes o patrones que permitan predecir el fraude a nivel transaccional, esto se refiere a encontrar comportamientos de compra extraños en base a la historia transaccional del cliente y datos de la transacción entrante. En esta memoria se aplicarán diversos modelos de detección sobre los datos disponibles para alcanzar este objetivo.

1.1. Motivación

Existen dos aristas que hacen este problema interesante. Primero, desde un punto de vista conceptual, es un problema de clasificación con ambigüedad, dos transacciones muy similares puede una ser fraudulenta y la otra no. No es claro cómo resolverlo, pues no hay leyes que describan el comportamiento de los clientes, por tratarse éste de un comportamiento social. Segundo, desde un punto de vista económico, es importante prevenir el fraude y dejar de perder dinero por este concepto, pues en una transacción normal la ganancia es un porcentaje de la venta, en cambio en una transacción fraudulenta la pérdida es casi total. Además como consecuencia se protege al cliente y le da mayor seguridad al momento de transaccionar con la tarjeta.

1.2. Alcances y Objetivo General

El objetivo principal es la detección de fraude a nivel transaccional en una tarjeta de crédito asociada a una empresa *Retail* (empresa dedicada a la venta al detalle). Para esto se comparan distintos modelos matemáticos y se ve cuál de ellos presenta mejor desempeño. Los modelos se construyen a nivel transaccional no a nivel cliente. Este trabajo contempla el desarrollo de los modelos, no así su implementación y posterior seguimiento y mantención. Esta memoria corresponde a un primer paso para una futura implementación de un modelo para poder prevenir el fraude.

1.3. Objetivos Específicos

Dentro de los objetivos específicos se encuentran:

- Identificación, extracción y preparación de datos.
- Definiciones asociadas al problema y transformación de datos.
- Selección de variables relevantes.
- Generación de modelos predictivos.
- Evaluación y comparación estadística de los modelos.
- Medición del impacto económico de los modelos.

1.4. Estructura de la memoria

La estructura utilizada en este documento para exponer el trabajo realizado es la siguiente:

- **Capítulo 1. Introducción:** corresponde a la descripción del tema, la motivación de éste y los alcances y objetivos del trabajo realizado.
- **Capítulo 2. Marco teórico:** se entrega el marco conceptual utilizado y la descripción del problema y los orígenes de datos.
- **Capítulo 3. Descripción de modelos:** se detallan los modelos usados para resolver el problema.
- **Capítulo 4. Tratamiento de datos:** corresponde al tratamiento de datos y selección de variables relevantes.
- **Capítulo 5. Entrenamiento y test:** se describe el entrenamiento y test de los modelos.

- **Capítulo 6. Resultados y análisis:** se entregan y analizan los resultados obtenidos.
- **Capítulo 7. Conclusiones:** se enumeran las conclusiones del trabajo realizado y se proponen trabajos a realizar en el futuro.

Capítulo 2

Marco teórico

2.1. Definiciones

Una transacción fraudulenta es definida en el negocio como aquellas transacciones que el cliente desconoce. Los expertos del negocio han definido una ventana de tiempo para conocer la totalidad de fraude de un mes dado. Esto genera un retraso en los datos, por ejemplo si la ventana es de cuatro meses, hay que esperar cuatro meses para conocer el fraude del mes presente.

2.2. Marco conceptual

El marco conceptual (o *Framework*) que será utilizado es el proceso conocido como KDD (*Knowledge Discovery in Database* o Descubrimiento de conocimiento en base de datos). "El KDD es el proceso no-trivial de identificar patrones previamente desconocidos, válidos, nuevos, potencialmente útiles y comprensibles dentro de los datos" [17]. El KDD es un proceso interactivo e iterativo, involucrando numerosos pasos con muchas decisiones hechas por el usuario. Así el KDD puede implicar importantes iteraciones y puede contener ciclos entre cualquier par de pasos. El flujo básico de pasos es mostrado en la figura 2.1. Este flujo está relacionado con los objetivos específicos definidos en 1.3 y es la metodología usada a lo largo de esta memoria.

La minería de datos (*Data Mining*) es un proceso en el que se puede encontrar información nueva y potencialmente útil en los datos. Existen diversos algoritmos utilizados en *Data Mining* y que han sido utilizados en la detección de fraude. Estos algoritmos se dividen en supervisados (o predictivos) y no supervisados (o del descubrimiento de conocimiento). En los primeros se sabe lo que se quiere encontrar, por ejemplo se puede tener una variable con una clasificación y se quiere predecir la clase de nuevos datos, en cambio en los segundos no se sabe esto con precisión. Así los algoritmos supervisados sirven para predecir (por ejemplo clasificar un nuevo dato), mientras que los no supervisados descubren patrones y tendencias en los datos. En esta memoria se usarán los primeros, pues se tiene una variable que dice si la transacción es normal o fraudulenta y se quiere predecir a que clase pertenece la transacción. Típicamente para los algoritmos supervisados se crea

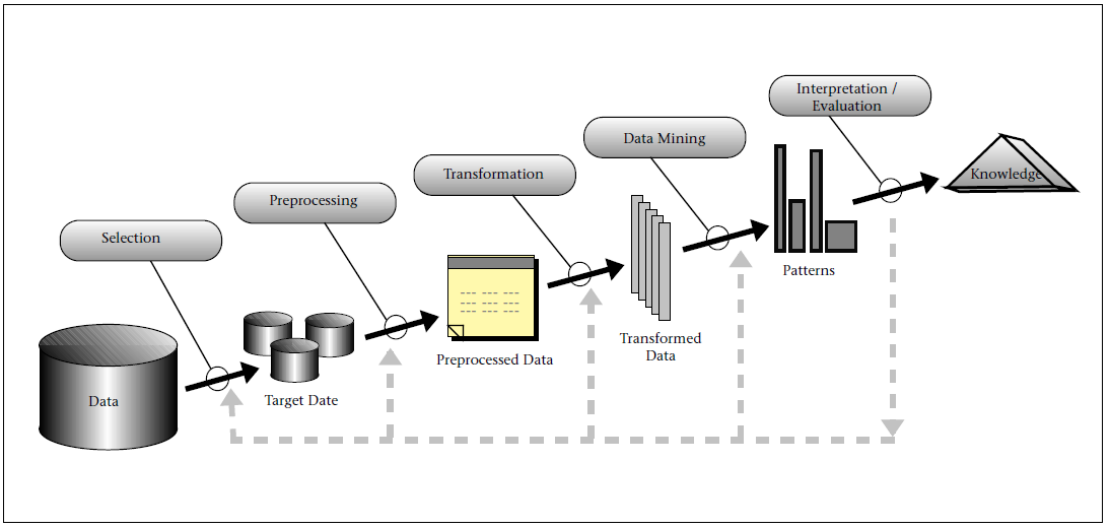

Figura 2.1: Esquema de los pasos que componen el proceso KDD.

una base de entrenamiento y otra de prueba o test, donde los modelos son creados o entrenados en la primera y son evaluados o puestos a prueba en la segunda.

2.3. Formulación del problema

Sea N el número de registros de la base de entrenamiento. Los datos en esta base son de la forma (x_i, Y_i), $i = 1, \ldots, N$, donde x_i es un vector con las variables de entrada, también llamadas variables explicativas o independientes, e Y_i es la variable objetivo o dependiente, codificada como 0 y 1, donde

$$Y_i = \begin{cases} 0 & \text{transacción normal} \\ 1 & \text{transacción fraudulenta} \end{cases} \tag{2.1}$$

Se busca una función $F(x_i)$ que diga si la transacción es normal o fraudulenta. La función F puede entregar valores en el intervalo $[0,1]$ (como una probabilidad) o puede tomar los valores 0 y 1 al igual que Y_i. Para mayor claridad, las clases a predecir se nombran al igual que en la literatura: positiva (P), si la transacción es fraudulenta y negativa (N) si no. Así para un clasificador se tienen cuatro posibilidades:

- Verdadero positivo (VP): la verdadera clase es positiva y se predice positiva.
- Falso negativo (FN): la verdadera clase es positiva y se predice negativa.
- Verdadero negativo (VN): la verdadera clase es negativa y se predice negativa.

		Clase verdadera	
		P	N
Clase predicha	P’	VP	FP
	N’	FN	VN

Tabla 2.1: Matriz de confusión.

- Falso positivo (FP): la verdadera clase es negativa y se predice positiva.

Estas cuatro salidas forman una matriz de 2x2, llamada matriz de confusión (o tabla de contingencia). En la tabla 2.1 se muestra un ejemplo de ésta. Con esta matriz se pueden generar diversas medidas para el rendimiento de los modelos predictivos. Para mayor información de estas medidas ver el apéndice A.

2.4. Dificultad del problema

Existen varias dificultades asociadas a la detección de fraude, tales como, clases desbalanceadas (*skewed data distributions*), escasez de datos reales, entre otros.

2.4.1. Clases desbalanceadas

La clase mayor, en este caso las transacciones normales, es muy superior en número a la clase menor (fraude). Esto es natural, pues para financiar una transacción fraudulenta se necesitan muchas transacciones normales, de lo contrario un comercio no es sustentable en el tiempo. La tasa de fraude que se tiene en este estudio es menor al 0,1 %, la cual no puede ser revelada por temas de confidencialidad. Para manejar clases desbalanceadas se proponen métodos como *undersampling* [3] y *oversampling* [36], que entregan una forma de elegir una muestra de casos apropiada para construir los modelos. El término *undersampling* se refiere a disminuir la proporción de la clase mayor, escogiendo un determinado número de casos de esta clase hasta alcanzar una distribución más balanceada con respecto a los casos escogidos de la clase menor. El término *oversampling* consiste en aumentar la proporción de la clase menor, repitiendo los casos de esta clase hasta alcanzar la distribución deseada con respecto a los casos escogidos de la clase mayor.

Otra consecuencia de tener tal desbalance de clases es que algunas medidas de rendimiento para los modelos no son muy útiles, por ejemplo se puede tener una precisión global (porcentaje de casos correctamente clasificados) cercana al 100 % clasificando todas las transacciones fraudulentas como normales.

2.4.2. Falta de datos reales

En la mayoría de las publicaciones sobre detección de fraude en tarjetas de crédito se habla de la falta de datos reales. De las revisadas para esta memoria unas cuantas tienen datos reales [2, 3, 10, 15, 26, 32, 45] y el resto usan datos sintéticos o simulados o datos de encuestas. La falta de datos se debe a que éstos son delicados y confidenciales, pues corresponden a las transacciones de grandes empresas y bancos, donde existe una gran competencia. Por esta razón, en las publicaciones que sí usan datos reales, tienen restricciones en la publicación, por ejemplo en el número de transacciones, montos, nombre de variables, etc. Estas situaciones hacen más difícil la búsqueda de referencias en la literatura. De las publicaciones sobre detección de fraude en tarjetas de crédito y que usan datos reales, sólo en [3] mencionan las variables usadas.

2.4.3. Dinámica del fraude

Otro problema es que los patrones de fraude son cambiantes en el tiempo. Cuando se toman medidas para evitar el fraude, al poco tiempo las personas que cometen fraude buscan maneras alternativas de evadir los controles, luego existe un ciclo entre los que hacen fraude y quienes lo desean detectar o prevenir, se da el "juego del gato y el ratón". Esto hace que detectar patrones de fraude sea muy difícil. Además los patrones de fraude dependen del tipo de transacción, ya que por ejemplo rubros como recargas telefónicas y pasajes aéreos tienen comportamiento distinto (montos, periodicidad, etc.). Para enfrentar el cambio en el tiempo se pueden tomar varios meses de transacciones y además enfocarse sólo en un tipo de transacción, para abordar la heterogeneidad de comportamiento.

También el mercado siempre está cambiando con la creación de nuevos comercios. Esta es otra fuente de ruido cada vez más importante. Por ejemplo en comercios nuevos, no se tiene una historia transaccional con la que se pueda comparar y poder detectar patrones extraños de compra.

2.5. Descripción de los orígenes de datos

Las transacciones se encuentran disponibles en el *Data WareHouse* (base de datos corporativa que se caracteriza por integrar y depurar información de una o más fuentes distintas), básicamente en tres bases: la propia, la no propia y la de transacciones financieras (TF). Existe esta distinción en las bases dada la situación particular de la empresa. Con la tarjeta de crédito se puede comprar en diversos comercios asociados, por nombrar algunos ejemplos, farmacias y supermercados, además se pueden hacer compras por internet y en el extranjero, luego se hace la distinción entre el comercio propio de la empresa y el resto. Además se tienen dos bases que contienen información de las transacciones fraudulentas.

2.5.1. Base propia

Por base propia se entienden las transacciones de compra de productos hechas sólo en el comercio propio, con cualquier medio de pago. Esta base no está a nivel transaccional, sino que posee un detalle mayor, pues cada registro o fila corresponde a un ítem o producto, no a una transacción propiamente tal, la cual se entiende a nivel boleta, es decir, una transacción una boleta.

2.5.2. Base no propia

En la base no propia se encuentran las transacciones hechas en el comercio no propio y con la tarjeta de crédito de la empresa. Típicamente en este tipo de comercios se acepta como medio de pago tarjetas asociadas a franquicias internacionales. Esta base se encuentra a nivel transaccional.

2.5.3. Base de transacciones financieras

Esta base contiene las transacciones financieras, que corresponden a giros de dinero en efectivo. Dada su naturaleza, esta base se encuentra a nivel transaccional.

2.5.4. Bases transacciones fraudulentas

Para las transacciones fraudulentas se tienen dos bases de datos, la base de fraude y una réplica del sistema de crédito.

Base de fraude

La base de fraude consiste en las transacciones reclamadas por los clientes, es decir, transacciones que un cliente desconoce. Esta base se genera mensualmente y se poseen más de 12 meses de historia. Contiene alrededor de 17 variables. Las primeras son para identificar la cuenta, por ejemplo: n° de cuenta, n° de tarjeta, etc. Existen ocho variables netamente de la transacción, dentro de ellas: código de rubro y comercio, fecha de transacción, monto, etc. Una variable de interés que no se tiene es el tipo de transacción.

Base réplica

La otra base, en adelante llamada la “base réplica”, se obtiene de un repositorio réplica, en el cual se encuentran todas las transacciones. Es típico generar réplicas, para poder extraer datos y hacer consultas sin afectar la operabilidad del sistema. Esta base tiene asociado un sistema que sirve para la detección de fraude, generando alertas para transacciones sospechosas. Con este sistema se pueden hacer consultas por las transacciones fraudulentas de cierto periodo, no así el total de transacciones. La base posee alrededor de 39 variables. Como apunta al sistema de crédito tiene

más información que la base de fraude, tal como, código de autorización, código de respuesta, etc. Dentro de las variables identificadoras se encuentran el n° de tarjeta y el n° de cuenta. Sobre la transacción se tienen variables tales como: fecha de transacción, código de comercio, monto, tipo de transacción, etc.

Capítulo 3

Descripción de modelos

En este capítulo se revisarán cuatro tipo de modelos que han sido aplicados en detección de fraude: redes neuronales artificiales [1, 32], *support vector machines* [3, 9], árboles de decisión [15, 36] y regresión logística [3]. Por esta razón estos modelos serán usados en esta memoria y porque son aplicables bajo los recursos disponibles.

3.1. Redes neuronales

3.1.1. Motivación biológica

El estudio de redes neuronales artificiales (ANN, *artifitial neural network*) ha sido inspirado en parte por la observación de que los sistemas biológicos de aprendizaje están construidos de complejas redes de neuronas interconectadas. La neurona es la unidad funcional fundamental de todo sistema nervioso, incluyendo el cerebro. Cada neurona consiste de un cuerpo celular, llamado soma, que contiene un núcleo. Del soma salen un número de fibras llamadas dendritas y una fibra larga llamada axón. Con estos elementos las neuronas se comunican con otras miles de neuronas haciendo sinapsis. En una ANN, nodos artificiales, también llamados neuronas, están conectados para formar una red de nodos que imitan una red neuronal biológica.

3.1.2. Redes neuronales artificiales

Una ANN es un modelo estadístico no lineal de clasificación o regresión de dos etapas, típicamente representado por un diagrama como en la figura 3.1. Esta red aplica para ambos, regresión y clasificación. Para regresión típicamente hay sólo un nodo output o salida Y_1. Para clasificación de K clases, hay K unidades de salida, con el k-ésimo nodo modelando la probabilidad de la clase k.

Una ANN consiste en un grupo de neuronas o nodos interconectados por enlaces o *links*.

Cada *link* tiene un valor numérico asociado, llamado peso. Los pesos son parámetros del modelo y el aprendizaje usualmente consiste en actualizar éstos.

Cada nodo tiene un conjunto de enlaces de entrada provenientes de otros nodos, un conjunto de enlaces de salida a otros nodos, un nivel de activación actual y una manera de calcular dicho nivel dependiendo de sus entradas y pesos. La idea es que cada nodo hace un cálculo local basado en sus entradas, pero sin un control global sobre el conjunto de nodos como un todo.

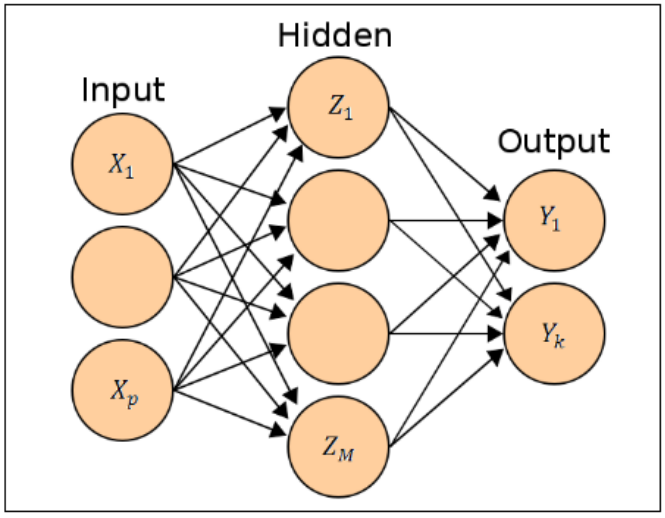

Figura 3.1: Esquema de una ANN *feed-forward* con una capa oculta.

Notación

I_i denota el nodo input o de entrada i-ésimo, cada uno de estos nodos almacena una variable de entrada. H_j denota el nodo j de una capa oculta (*hidden layer*), en estos nodos se hacen gran parte de los cálculos. O_k denota el nodo output k-ésimo. W_{ji} es el peso en el enlace o *link* desde el nodo j al i. a_i es el valor de activación del nodo i (también corresponde a la salida del nodo) y g es la función de activación.

Nodo

En la figura 3.2 se muestra un típico nodo. Cada nodo realiza el siguiente cálculo: recibe una señal de sus enlaces de entrada y calcula un nuevo nivel de activación, el cual envía a través de sus enlaces de salida. El cálculo del nivel de activación está basado en los valores de cada señal de entrada recibida de un nodo vecino y los pesos en cada enlace de entrada. Este cálculo está separado en dos componentes: 1) una componente lineal, llamada función de entrada, in_i, que calcula la suma ponderada de los valores de entrada y 2) una componente no-lineal, llamada función de activación, g, que transforma la suma ponderada en el valor final que sirve como valor de activación, a_i.

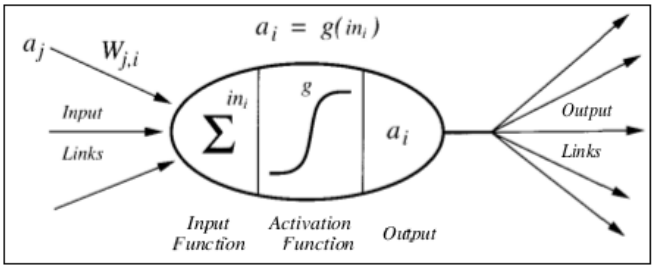

Figura 3.2: Diagrama de un nodo.

La entrada total de un nodo es la suma de los valores de activación de entrada, ponderados por sus respectivos pesos:

$$in_i = \sum_j W_{ji} a_j \tag{3.1}$$

Cada nodo calcula su nuevo valor de activación aplicando la función de activación, g, al resultado de la función de entrada:

$$a_i = g\left(\sum_j W_{ji} a_j\right) \tag{3.2}$$

Diferentes modelos son obtenidos usando diferentes funciones matemáticas para g. Tres elecciones comunes son las funciones *Step*, *Sign* y *Sigmoid* (funciones escalera, signo y sigmoide, respectivamente), ilustradas en la figura 3.3.

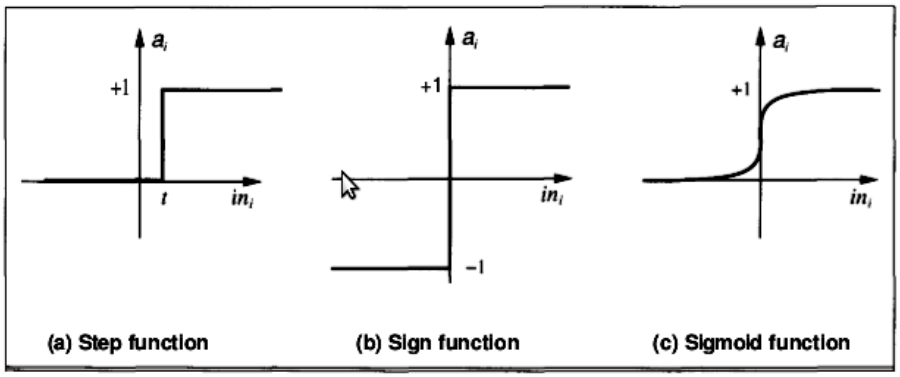

Figura 3.3: Tres diferentes funciones de activación para los nodos.

Hay una variedad de estructuras para redes neuronales, cada una con diferentes propiedades computacionales. La principal distinción a ser hecha es entre redes *feed-forward* y recurrentes. En

una red *feed-forward* los enlaces son unidireccionales y no hay ciclos. En una red recurrente los enlaces pueden formar arbitrarias topologías. Técnicamente hablando, una red *feed-forward* es un grafo acíclico dirigido. Usualmente las redes neuronales son organizadas en capas. En estas redes cada nodo está conectado solo con nodos en las capas vecinas, no hay enlaces entre los nodos de una misma capa. En la figura 3.4 se muestra un simple ejemplo de una red neuronal *feed-forward* con capas. Esta red tiene dos capas, pues los nodos de entrada (los nodos cuadrados) sirven sólo para pasar la activación a la siguiente capa, no son contados. El valor de activación de cada nodo de entrada es determinado externamente (variables de entrada). A la derecha de la red se tiene un nodo de salida, el cual entrega la salida del modelo. Entre los nodos de entrada y salida se encuentran nodos que no tienen conexión directa con el medio externo. Estos nodos son llamados nodos ocultos, porque no pueden ser observados directamente por el comportamiento de entrada y salida de la red neuronal. Las redes que no poseen nodos ocultos se llaman perceptrones, en éstos el aprendizaje es mucho más simple, pero son muy limitados en cuanto a lo que pueden representar. Redes con una o más capas ocultas son llamadas redes multicapa.

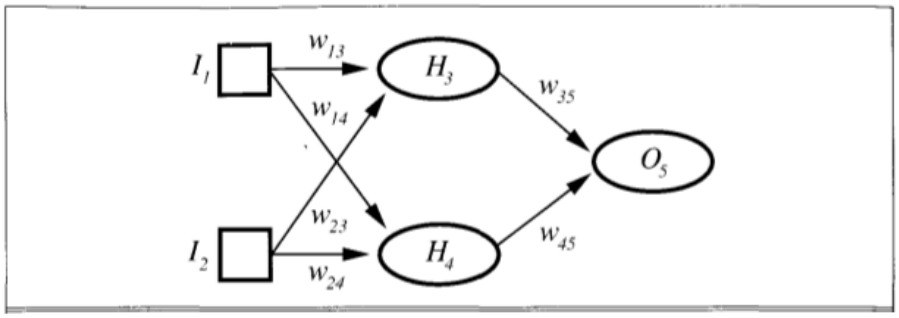

Figura 3.4: Red neuronal *feed-forward* de dos capas, con dos nodos de entrada, dos nodos ocultos y un nodo de salida.

En esta memoria se usarán redes neuronales *feed-forward* multicapa porque son las más usadas y estudiadas.

Perceptrón

Redes neuronales *feed-forward* con capas fueron estudiadas a finales de la década de 1950 bajo el nombre de perceptrones. Aunque redes de todos los tamaños y topologías fueron consideradas, el único aprendizaje efectivo en ese tiempo fue para redes neuronales de una capa, entonces los esfuerzos fueron puestos en este tipo de redes. Hoy en día el nombre de perceptrón es usado como sinónimo para un red neuronal *feed-forward* de una capa.

Si se denota el nodo de salida como O y los pesos desde el nodo j a O como W_j, la activación del nodo de entrada j es dado por I_j y la activación del nodo de salida es por tanto:

$$O = func\left(\sum_j W_j I_j\right) \tag{3.3}$$

Dentro de las limitaciones de un perceptrón es que una función puede ser representada por uno si y solo si ésta es linealmente separable. A continuación se describe el método de aprendizaje de un perceptrón: la red inicial tiene asignados pesos aleatorios, usualmente entre -0,5 y 0,5. La red es entonces actualizada para tratar de ser consistente con los casos de entrenamiento, esto se hace con pequeños ajustes en los pesos para reducir la diferencia entre lo observado y lo predicho. Es necesario repetir esta fase de actualización varias veces para cada caso para obtener convergencia. Típicamente, el proceso de actualización es dividido en épocas y en cada época se actualizan todos los pesos de todos los casos. En la figura 3.5 se muestra el método general de aprendizaje en una ANN: ajustar los pesos hasta que los valores output y los valores reales concuerden.

```
function ANN-Learning(casos) returns network
    network ← una red con pesos asignados aleatoriamente
    while(algún caso no es predicho correctamente o no se alcanza el criterio de parada)
        for(c en casos)
            O ← ANN-Output(network,c)
            T ← el valor de salida observado a partir de c
            actualizar los pesos en red basado en c, O y T
        end
    return network
```

Figura 3.5: Método general de aprendizaje en una ANN.

Para perceptrones, la regla de actualización de los pesos es bastante simple, si la salida predicha para un nodo de salida es O y la salida correcta debiera ser T, entonces el error es dado por:

$$Err = T - O \tag{3.4}$$

Si el error es positivo, entonces es necesario incrementar O, si éste es negativo es necesario decrecer O. Ahora, cada nodo de entrada contribuye con $W_j I_j$ a la entrada total, entonces si I_j es positivo, un incremento en W_j tiende a incrementar O y si I_j es negativo, un incremento en W_j tiende a decrecer O. Así, se puede lograr el efecto que se desea con la siguiente regla,

$$W_j \leftarrow W_j + \alpha \times I_j \times Err \tag{3.5}$$

donde el término α es una constante llamada tasa de aprendizaje. Esta regla es una pequeña variación de la regla de aprendizaje del perceptrón propuesta por Frank Rosenblatt en 1960 [40]. Rosenblatt probó que un sistema de aprendizaje usando esta regla de aprendizaje converge a un conjuntos de pesos que representan correctamente a los casos de entrenamiento, mientras los casos representen una función linealmente separable.

Redes neuronales feed-forward multicapa

El método más popular para aprendizaje en redes neuronales multicapa es llamado *back-propagation*. Éste fue inventado en 1969 por Bryson y Ho [7], pero fue más o menos ignorado hasta mediados de los 80's. Las razones para esto pueden ser sociológicas, pero puede también tener relación con los requerimientos computacionales del algoritmo en problemas no triviales. El aprendizaje en estas redes procede de la misma forma que en perceptrones: los casos de entrenamiento son presentados a la red y si la red calcula una salida que es igual al objetivo nada se hace, si hay un error (una diferencia entre la salida y el objetivo) entonces los pesos son ajustados para reducir el error. En perceptrones esto es fácil, porque sólo hay un peso entre cada entrada y salida. Pero en redes neuronales multicapa hay muchos pesos conectando cada entrada a una salida y cada uno de esos pesos contribuye a más de una salida.

El algoritmo *back-propagation* es un enfoque sensible para dividir la contribución de cada peso. Si Err_i es el error $(T_i - O_i)$ en el nodo de salida, entones la regla de actualización de pesos para el enlace desde el nodo j al i es :

$$W_{ji} \leftarrow W_{ji} + \alpha \times a_j \times Err_i \times g'(in_i)$$

donde g' es la derivada de la función g. Se define un nuevo término de error Δ_i, el cual para el nodo de salida se define como $\Delta_i = Err_i \times g'(in_i)$. La regla queda como:

$$W_{ji} \leftarrow W_{ji} + \alpha \times a_j \times \Delta_i \tag{3.6}$$

Para actualizar las conexiones de los nodos ocultos la idea es que un nodo oculto j es "responsable" por alguna fracción del error Δ_i de cada nodo salida al que está conectado. Así, los valores Δ_i son divididos de acuerdo a la fuerza de las conexiones entre el nodo oculto y el nodo de salida y propagado hacia atrás para obtener los valores Δ_j de las capas ocultas. La regla de propagación es:

$$\Delta_j = g'(in_j) / \sum_i W_{ji}\Delta_i \tag{3.7}$$

Ahora la regla de actualización de pesos entre la capa de entrada y las capas ocultas es:

$$W_{kj} \leftarrow W_{kj} + \alpha \times a_k \times \Delta_j \tag{3.8}$$

El detalle del algoritmo es mostrado en la figura 3.6.

Back-propagation puede ser visto como un método de descenso del gradiente en el espacio de los pesos. En este caso, el gradiente está en la superficie de error: la superficie que describe el

function Back-Prop-Update(*network*,*casos*,α) **returns** *network* con pesos modificados
inputs: *network*, una red multicapa
casos, muestra de entrenamiento
α, la tasa de aprendizaje

while(la red no converja)
for(*c* en *casos*)
* *Calcular la salida para este caso* *\
$O \leftarrow$ RUN-Network(*network*,*c*)
* *Calcular el error y* Δ *para los nodos en la capa de salida* *\
Err $\leftarrow T$-O
* *Actualizar los pesos hacia la capa de salida* *\
$W_{ji} \leftarrow W_{ji} + \alpha \times a_j \times Err_i \times g'(in_i)$
for(subsecuente capa en *network*)
* *Calcular el error en cada nodo* *\
$\Delta_j = g'(in_j) / \sum_i W_{ji}\Delta_i$
* *Actualizar los pesos en la capa* *\
$W_{kj} \leftarrow W_{kj} + \alpha \times a_k \times \Delta_j$
end
end
return *network*

Figura 3.6: Algoritmo back-propagation para actualizar los pesos en una red multicapa.

error en cada caso de entrenamiento como función de todos los pesos de la red. *Back-propagation* da una manera de dividir el cálculo del gradiente a través de los nodos, así el cambio en cada peso puede ser calculado por el nodo al cual el peso está vinculado, usando sólo información local.

A continuación se derivan las ecuaciones de *Back-propagation* desde el principio. Para la función de error se usa la suma de los errores cuadráticos sobre los valores de salida:

$$E = \frac{1}{2}\sum_i (T_i - O_i)^2 \tag{3.9}$$

La clave es que los valores de salida O_i son funciones de los pesos (como en la ecuación (3.3)). Para una red neuronal de dos capas se tiene:

$$\begin{aligned} E(W) &= \frac{1}{2}\sum_i \left(T_i - g\left(\sum_j W_{ji} a_j\right)\right)^2 \\ &= \frac{1}{2}\sum_i \left(T_i - g\left(\sum_j W_{ji} g\left(\sum_k W_{kj} I_k\right)\right)\right)^2 \end{aligned} \tag{3.10}$$

Aunque el término a_j es una expresión compleja, no depende de W_{ji}. Además, sólo uno de los términos en la sumatoria sobre i y j depende en un particular W_{ji}, entonces todos los otros términos son tratados como constantes con respecto a W_{ji} y desaparecen al diferenciar. Luego al diferenciar la primera linea con respecto a W_{ji} se obtiene:

$$\begin{aligned} \frac{\partial E}{\partial W_{ji}} &= -a_j(T_i - O_i)\left(\sum_j W_{ji}\right) \\ &= -a_j(T_i - O_i)g'(in_i) = -a_j\Delta_i \end{aligned} \tag{3.11}$$

La derivada del gradiente con respecto a W_{kj} es un poco más compleja, pero entrega un resultado similar:

$$\frac{\partial E}{\partial W_{kj}} = -I_k\Delta_j \tag{3.12}$$

Para obtener las reglas de actualización de los pesos, se tiene que recordar que el objetivo es minimizar el error, luego se tienen que tomar pequeños pasos en la dirección opuesta al gradiente.

3.2. Support Vector Machines

3.2.1. Introducción

Support vector machine (SVM) es una técnica usada en clasificación basada en encontrar un hiperplano de separación que divida el espacio de entrada en dos regiones. Este hiperplano produce una frontera no lineal que es construida a partir de una frontera lineal en un espacio transformado y de mayor dimensión que el espacio de entrada. El algoritmo original de SVM fue inventado por Vladimir N. Vapnik en 1963 y la versión estándar actual fue propuesta por Vapnik y Corinna Cortes en 1995 [11]. Ésta extiende a las SVMs al caso de clases que no son linealmente separables mediante la introducción de las llamadas funciones de *kernel* o núcleo y de variables de pérdida o *slacks*.

3.2.2. Formulación matemática

Sea N el tamaño del conjunto de entrenamiento y p la cantidad de variables. Cada dato es representado por un par (x_i, y_i), donde $x_i \in \mathbb{R}^p$, $y_i \in \{-1, 1\}$, $i = 1, \ldots, N$. i es el índice del caso, x_i es un vector con las variables del caso i e y_i muestra a que clase pertenece. Se define un hiperplano como,

$$\{x : f(x) = x^T\beta + \beta_0 = 0\} \tag{3.13}$$

donde β es un vector unitario y $\beta_0 \in \mathbb{R}$. Así una regla de clasificación inducida por $f(x)$ es:

$$G(x) = sign\left[x^T\beta + \beta_0\right] \tag{3.14}$$

Si las clases son separables, se puede encontrar una función $f(x) = x^T\beta + \beta_0$, tal que $y_i f(x_i) > 0, \forall i$. Se quiere encontrar el hiperplano con el margen más grande entre los puntos de las clases 1 y -1. Luego el problema de optimización es,

$$\begin{array}{c} \text{máx}_{\beta,\beta_0,||\beta||=1}\, C \\ \text{s.a. } y_i\left(x_i^T\beta + \beta_0\right) \geq C, i = 1,\ldots,N \end{array} \tag{3.15}$$

donde C es el margen (distancia entre el hiperplano y el punto más cercano de cada clase a éste). Este problema es equivalente a,

$$\begin{array}{c} \text{mín}_{\beta,\beta_0}\, ||\beta|| \\ \text{s.a. } y_i\left(x_i^T\beta + \beta_0\right) \geq 1, i = 1,\ldots,N \end{array} \tag{3.16}$$

donde se ha quitado la restricción en la norma de β. Notar que $C = 1/||\beta||$. Si las clases no son linealmente separables, entonces existe un solape entre las clases. Se puede maximizar C, pero permitiendo que algunos puntos estén en la clase incorrecta. Sea $\xi = (\xi_1, \xi_2, \ldots, \xi_N)$ las variables *slacks* (permiten relajar el problema), entonces el problema es:

$$\begin{array}{c} \text{mín}_{\beta,\beta_0}\, ||\beta|| \\ \text{s.a. } y_i\left(x_i^T\beta + \beta_0\right) \geq 1 - \xi_i, \forall i \\ \xi_i \geq 0, \sum \xi_i \leq constante \end{array} \tag{3.17}$$

El valor ξ_i en la primera restricción es la cantidad proporcional por la cual la predicción está en el lado incorrecto de su margen. Acotar la suma de los ξ_i es acotar las clasificaciones malas (ocurren cuando $\xi_i > 1$). Así se tiene un problema cuadrático con restricciones lineales, por tanto es un problema de optimización convexo. El problema anterior es equivalente a,

$$\begin{array}{c} \text{mín}_{\beta,\beta_0}\, \frac{1}{2}||\beta|| + \gamma\sum_{i=1}^N \xi_i \\ \text{s.a. } \xi_i \geq 0, y_i\left(x_i^T\beta + \beta_0\right) \geq 1 - \xi_i, \forall i \end{array} \tag{3.18}$$

donde γ reemplaza la constante, el caso separable es $\gamma = \infty$. La función de Lagrange (primal) es:

$$L_p = \frac{1}{2}||\beta||^2 + \gamma \sum_{i=1}^{N} \xi_i - \sum_{i=1}^{N} \alpha_i \left[y_i \left(x_i^T \beta + \beta_0 \right) - (1 - \xi_i) \right] - \sum_{i=1}^{N} \mu_i \xi_i \tag{3.19}$$

Derivando c/r a β, β_0, ξ_i e igualando a cero se obtiene,

$$\beta = \sum_{i=1}^{N} \alpha_i y_i x_i \tag{3.20}$$

$$0 = \sum_{i=1}^{N} \alpha_i y_i \tag{3.21}$$

$$\alpha_i = \gamma - \mu_i, \forall i \tag{3.22}$$

y las restricciones de positividad α_i, μ_i , $\xi_i \geq 0$. Al reemplazar (3.20)-(3.22) en (3.19) se obtiene el Lagrangiano de la función objetivo dual (*Wolfe*),

$$L_D = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{i'=1}^{N} \alpha_i \alpha_{i'} y_i y_{i'} x_i^T x_{i'} \tag{3.23}$$

el cual da una cota inferior a la función objetivo en (3.18) para cualquier punto factible. Se maximiza L_D s.a. $0 \leq \alpha_i \leq \gamma$ y (3.21). Además se tienen las condiciones de KKT que añaden las restricciones,

$$\alpha_i \left[y_i \left(x_i^T \beta + \beta_0 \right) - (1 - \xi_i) \right] = 0 \tag{3.24}$$

$$\mu_i \xi_i = 0 \tag{3.25}$$

$$y_i \left(x_i^T \beta + \beta_0 \right) - (1 - \xi_i) \geq 0 \tag{3.26}$$

para $i = 1, \ldots, N$. Las ecuaciones (3.20)-(3.26) caracterizan únicamente la solución del problema primal y dual. De (3.20) podemos ver una solución para β de la forma,

$$\hat{\beta} = \sum_{i=1}^{N} \hat{\alpha}_i y_i x_i \tag{3.27}$$

con coeficientes $\hat{\alpha}_i$ no ceros sólo para las observaciones i para las cuales (3.26) se tiene con igualdad (debido a (3.24)). Estas observaciones son llamadas vectores soportes (support vectors).

Entre éstos, algunos estarán en el margen $\left(\hat{\xi}_i = 0\right)$ y de (3.22) y (3.25) serán caracterizados por $(0 < \hat{\alpha}_i < \gamma)$. Para el resto $\left(\hat{\xi}_i > 0\right)$ se tiene $\hat{\alpha}_i = \gamma$. Luego de (3.24) los puntos en el margen pueden ser usados para obtener β_0. Maximizar (3.23) es un problema de programación cuadrática convexa más simple que el primal (3.19), y puede ser resuelto con técnicas estándares. Dado $\hat{\beta}$ y $\hat{\beta}_0$ la función de decisión es:

$$G(x) = sign\left[\hat{f}(x)\right] = sign\left[x^T\hat{\beta} + \hat{\beta}_0\right] \tag{3.28}$$

El parámetro de ajuste es el parámetro de costo γ.

Se puede hacer el procedimiento más flexible agrandando el espacio usando expansiones bases, tal como polinomios o *splines*. Sean las funciones bases $h_m(x)$, $m = 1,\ldots,M$. Se procede igual que antes usando como entrada $h(x_i) = (h_1(x_i), h_2(x_i), \ldots, h_M(x_i)), i = 1,\ldots,N$ y se obtiene la función (no lineal) $\hat{f}(x) = h(x)^T\hat{\beta} + \hat{\beta}_0$ y $\hat{G}(x) = sign\left(\hat{f}(x)\right)$ como antes. Se puede representar (3.19) y su solución de una forma especial que sólo involucra el producto interno de las entradas. Ahora el Lagrangiano de la función dual (3.23) tiene la forma:

$$L_D = \sum_{i=1}^{N}\alpha_i - \frac{1}{2}\sum_{i=1}^{N}\sum_{i'=1}^{N}\alpha_i\alpha_{i'}y_iy_{i'} < h(x_i), h(x_{i'}) > \tag{3.29}$$

De (3.20) la función de solución $f(x)$ puede ser escrita como:

$$\begin{aligned} f(x) &= h(x)^T\beta + \beta_0 \\ &= \sum_{i=0}^{N}\alpha_i y_i < h(x), h(x_i) > +\beta_0 \end{aligned} \tag{3.30}$$

Como antes, α_0 y β_0 pueden ser determinados resolviendo $y_i f(x_i) = 1$ en (3.30) para todos los x_i para los cuales $0 < \hat{\alpha}_i < \gamma$.

Así (3.29) y (3.30) dependen de $h(x)$ sólo a través del producto interno. De hecho, no es necesario especificar las transformaciones $h(x)$, sino solamente la función *kernel* o de núcleo,

$$K(x,x') = < h(x), h(x') > \tag{3.31}$$

que calcula el producto interno en el espacio transformado. Tres populares elecciones para K en la literatura de SVM son:

$$\begin{aligned} \text{Polinomio de grado d} &: K(x,x') = (1+ < x,x' >)^d \\ \text{Base radial} &: K(x,x') = exp(-\eta||x-x'||^2) \\ \text{Red neuronal} &: K(x,x') = \tanh(\kappa_1 < x,x' > +\kappa_2) \end{aligned}$$

Luego, por (3.30) se puede escribir la función como:

$$\hat{f}(x) = \sum_{i=1}^{N} \hat{\alpha}_i y_i K(x,x_i) + \hat{\beta}_0 \tag{3.32}$$

3.3. Árboles de decisión

3.3.1. Introducción

Los árboles de decisión son herramientas que usan un grafo tipo árbol o un modelo de decisión y sus posibles consecuencias. Éstos son una manera de visualizar un algoritmo. Los árboles de decisión son comúnmente usados en estadística, *data mining* y máquinas de aprendizaje como modelos predictivos, porque son fáciles de entender e interpretar. El objetivo es crear un modelo que prediga el valor de una variable objetivo, basado en varias variables de entrada. A continuación se muestran elementos comunes a los árboles de decisión y se verán dos tipos de éstos: C4.5 y CART.

Estructura

Las entradas son los casos de entrenamiento, cada uno correspondiendo a una tupla de atributos fijos (variables independientes) $A = \{A_1, A_2, \ldots, A_p\}$ y una variable C con la clasificación del caso (variable dependiente). Los atributos pueden ser continuos o discretos y la variable dependiente C es discreta. El objetivo es aprender de los casos de entrenamiento una función,

$$DOM(A_1) \times DOM(A_2) \times \ldots \times DOM(A_p) \rightarrow DOM(C) \tag{3.33}$$

que mapea desde los valores de los atributos a la clase predicha. Los árboles de decisión pueden ser vistos de una forma recursiva:

- un nodo hoja con una clase asociada o
- un nodo test que tiene dos o más salidas, cada una enlazada a un subárbol.

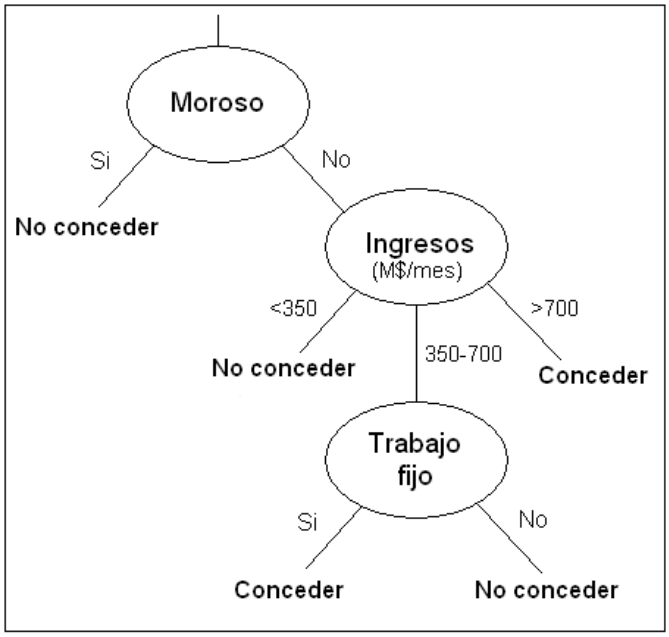

Figura 3.7: Ejemplo simple de un árbol de decisión para conceder un crédito.

En la figura 3.7 se muestra un ejemplo de un árbol de decisión.

Para clasificar un caso usando un árbol se comienza por la raíz y se sigue de la sgte. manera:

- si el nodo es un nodo hoja, la clase asociada a la hoja se convierte en la clase predicha,
- si el nodo es un nodo test, la salida de ese test es determinado y se sigue al nodo raíz del subárbol para esa salida.

Dividir y conquistar

Los árboles de decisión usan un método llamado "dividir y conquistar" para construir un árbol adecuado a partir de un conjunto de entrenamiento S:

- Si todos los casos en S pertenecen a la misma clase, el árbol de decisión pertenece a esa clase.
- En otro caso, sea B algún test con salidas $b_1, b_2, \ldots, b_t$ que producen una partición no trivial de S y S_i denota los casos en S que tienen salida b_i de B. El árbol de decisión se muestra en la figura 3.8. T_i es el árbol de decisión de repetir lo anterior para los casos en S_i.

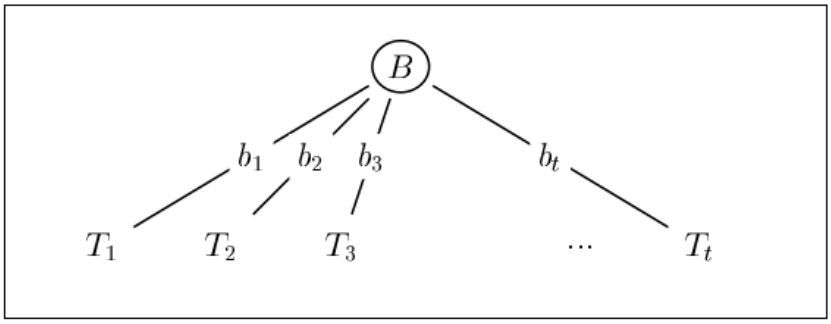

Figura 3.8: Árbol construido por el método dividir y conquistar.

3.3.2. C4.5

C4.5 pertenece a una sucesión de árboles de decisión que tiene sus orígenes en el trabajo de Hunt y otros a finales de los 50's y comienzos de los 60's [24]. Su predecesor inmediato fue ID3 [38] y aunque C4.5 ha sido sucedido por C5.0, éste último siendo una versión comercial, se pondrá enfoque en C4.5 dado que su código fuente está disponible.

Nodos Tests

C4.5 usa tests de tres tipos y cada uno involucra sólo un atributo A_i. Las regiones de decisión están así acotadas por hiperplanos, cada uno ortogonal a un eje de atributo.

- Si A_i es un atributo discreto con z valores, los posibles tests son:
  - "$A_i = ?$" con z salidas, una por cada valor de A_i.
  - "$A_i \in G_?$" con $2 \leq g \leq z$ salidas, donde $G = \{G_1, G_2, \ldots, G_g\}$ es una partición de los valores del atributo A_i. Tests de este tipo son encontrados por una búsqueda *greedy*[1] de una partición G que maximice el valor del criterio de *splitting* (discutido más abajo).
- Si A_i tiene valores numéricos, la forma del test es "$A_i \leq \theta$" con salida verdadero o falso, donde θ es una constante de umbral. Posibles valores de θ son encontrados ordenando los distintos valores de A_i que aparecen en S e identificando un umbral entre cada par de valores adyacentes (si por ejemplo S tiene d distintos valores para A_i, $d-1$ umbrales son considerados).

Criterios de splitting

En el algoritmo dividir y conquistar, cualquier test B que particione S no trivialmente genera un árbol de decisión, pero diferentes B's generan diferentes árboles. La mayoría de los sistemas

[1] Una búsqueda *greedy* sigue el problema de resolución de una heurística eligiendo el óptimo local en cada etapa con la esperanza de encontrar el óptimo global.

de aprendizaje intentan mantener el árbol pequeño como sea posible, porque son más fáciles de entender y, por "Occam's Razor arguments", son más probables de tener alta precisión predictiva [39]. Como es infactible garantizar la minimalidad del árbol [25], C4.5 usa búsqueda *greedy*[1], seleccionando el candidato test que maximiza la heurística *splitting criterion*.

Dos criterios son usados en C4.5, *information gain* (ganancia de información) y *gain ratio* (razón de ganancia). Sea $FR(C_j, S)$ la frecuencia relativa de casos en S que pertenecen a la clase C_j. La información de un mensaje que identifica la clase de un caso en S es definida como:

$$I(S) = -\sum_{j=i}^{k} FR(C_j, S) \log(FR(C_j, S)) \tag{3.34}$$

Después que S es particionado en subconjuntos $S_1, S_2, \ldots, S_t$ por un test B, la información ganada es:

$$G(S,B) = I(S) - \sum_{i=1}^{t} \frac{|S_i|}{|S|} I(S_i) \tag{3.35}$$

El criterio de ganancia elige el test B que maximice $G(S,B)$. Un problema con este criterio es que favorece los tests con numerosas salidas, por ejemplo, $G(S,B)$ es maximizada por un test en el cual cada S_i contiene sólo un caso. El criterio de razón de ganancia evita este problema tomando también en cuenta la información potencial de la partición misma:

$$P(S,B) = -\sum_{i=1}^{t} \frac{|S_i|}{|S|} \log\left(\frac{|S_i|}{|S|}\right) \tag{3.36}$$

La razón de ganancia elige, de los tests con ganancia mayor o igual al promedio, el test B que maximiza $G(S,B)/P(S,B)$.

Evitando overfitting

El algoritmo dividir y conquistar particiona los datos hasta que cada hoja contiene casos de sólo una clase o hasta que no es posible particionar más porque dos casos tienen los mismos valores para cada atributo, pero pertenecen a diferentes clases. Consecuentemente, si no hay casos conflictivos, el árbol de decisión clasificará correctamente todos los casos de entrenamiento. Esto es llamado *overfitting* (el modelo se sobreajusta a los datos de entrenamiento) y generalmente genera pérdida en la precisión predictiva en la mayoría de las aplicaciones. El *overfitting* puede ser evitado por un criterio de parada que previene que algunos conjuntos de casos de entrenamiento sean subdivididos o removiendo alguna parte de la estructura del árbol de decisión después que éste es producido.

Estimando las tasas de error verdadera

Sea Z algún clasificador formado a partir de un conjunto S de entrenamiento que clasifica mal M de los casos en S. La tasa de error verdadero de Z es la precisión sobre todo el universo desde el cual el entrenamiento fue obtenido. La tasa de error verdadera es usualmente más alta que la tasa de error de clasificación en el entrenamiento ($M/|S|$), la cual puede ser cercana a cero para árboles no podados. C4.5 estima la tasa de error verdadera de Z usando sólo los valores M y $|S|$ del conjunto de entrenamiento como sigue.

Si un evento ocurre M veces en N pruebas, la razón M/N es un estimador de la probabilidad p del evento. Se puede derivar una confianza límite para p; para una confianza CF, un límite superior p_r se puede encontrar tal que $p \leq p_r$ con probabilidad $1-CF$. p_r satisface las siguientes ecuaciones [12]:

$$CF = \begin{cases} (1-p_r)^N & \text{para } M = 0 \\ \sum_{i=0}^{M} \binom{N}{i} p_r^i (1-p_r)^{(N-i)} & \text{para } M > 0 \end{cases} \tag{3.37}$$

Ahora, Z puede ser visto como causante de M errores en $|S|$ pruebas. Como S fue construido para ajustar los casos en S y tiende a minimizar la tasa de error, p_r es usado como una estimación más conservativa de la tasa de error de Z en los casos no vistos. En lo que sigue $U_{CF}(M,N)$ denotará la cota de error p_r. C4.5 usa un CF de $0,25$, pero puede ser alterado para causar niveles altos o bajos de poda.

Método de poda

Después de que un árbol de decisión es producido por el algoritmo de dividir y conquistar, C4.5 poda éste en una simple pasada de abajo hacia arriba. Sea T un árbol de decisión (sin ser sólo una hoja), producido por un conjunto de entrenamiento S, como en la figura 3.9, donde cada T_i^* ya ha sido podado.

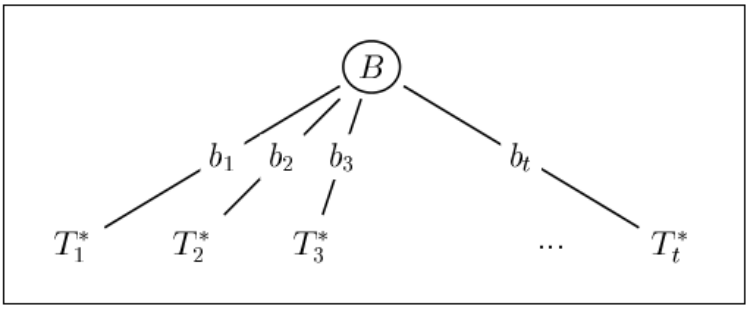

Figura 3.9: Esquema de árbol para poda.

Además, sea T_f^* el subárbol correspondiente a la salida más frecuente de B y sea L una hoja asociada a la clase más frecuente en S. Por último, sean E_T, $E_{T_f^*}$ y E_L los números de casos en S

mal clasificados por T, T_f^* y L, respectivamente. El algoritmo de poda de C4.5 considera tres tasas estimadas de errores:

- $U_{CF}(E_T, |S|)$,
- $U_{CF}(E_L, |S|)$, y
- $U_{CF}(E_{T_f^*}, |S|)$.

Dependiendo de cual es menor, C4.5:

- deja T sin cambios,
- reemplaza T por la hoja L, o
- reemplaza T por el subárbol T_f^*.

Esta forma de poda es computacionalmente eficiente y da resultados razonables en la mayoría de las aplicaciones.

3.3.3. CART

CART son las siglas para *Classification And Regression Trees* (árboles de regresión y clasificación) y es descrito en [6]. El uso de árboles en la comunidad estadística data de AID (*Automatic Interaction Detection*) por Morgan y Sonquist [33] y el posterior trabajo en THAID (*THeta Automatic Interaction Detection*) por Morgan y Messenger a principios de los 70's.

Nodos Tests

CART construye árboles con nodos tests sólo con salidas binarias. Si la clase es binaria, la restricción de división binaria permite a CART óptimamente particionar atributos categóricos (minimizando cualquier criterio de división cóncavo) a dos subconjuntos de valores en un tiempo lineal en el número de atributos. Aunque esta restricción tiene desventajas: quizás divisiones binarias no son buenas en un atributo que tiene una buena multi-división, lo cual podría guiar a árboles más pequeños.

Criterios de splitting

CART usa el índice de diversidad de Gini como criterio de división. Al igual que antes $FR(C_j, S)$ denota la frecuencia relativa de casos en S que pertenece a la clase C_j. El índice de Gini es definido como,

$$I_{gini}(S) = 1 - \sum_{j=1}^{k} FR(C_j, S)^2 \tag{3.38}$$

y la ganancia de información debido a la división es calculada como en la ecuación (3.35). Un árbol de probabilidades de clase predice una distribución de clase para un caso en vez de una simple clase. Una medida usual del rendimiento de un árbol de probabilidades de clase es el error cuadrático medio (ECM). Para cada clase j, sea $I_j(c)$ una variable indicadora, que es 1 si la clase para el caso c es j y 0 si no. El ECM es definido como,

$$ECM = E_c \left[\sum_{j=1}^{k} (I_j(c) - P_j(c))^2 \right] \tag{3.39}$$

donde la esperanza es sobre todos los casos y $P_j(c)$ representa la probabilidad asignada a la clase j para el caso c por el clasificador probabilístico. El índice de Gini minimiza la resubstitución estimada por el error cuadrático medio (el error de resubstitución es la tasa de error en los datos de entrenamiento).

Método de poda

CART usa una técnica de poda llamada *minimal cost complexity pruning*, el cual asume que el sesgo en el error de resubstitución de un árbol crece linealmente con el número de nodos hojas. El costo asociado a un subárbol es la suma de dos términos: el error de resubstitución y el número de hojas por un parámetro de complejidad α. Formalmente,

$$R_\alpha = R(T) + \alpha N_{hojas} \tag{3.40}$$

donde N_{hojas} es el número de hojas. Se puede mostrar que para cada valor de α existe un único árbol más pequeño que minimiza R_α [6]. Aunque α puede tomar valores reales, existe a lo más un número finito de subárboles posibles. Así hay una secuencia de árboles que minimizan R_α, $T_1 \succ T_2 \succ \ldots \succ T_t$, creada variando α de cero a infinito. Cada árbol está contenido en el árbol previo.

Cuando hay suficientes datos disponibles, seleccionar el mejor α para minimizar el error verdadero puede ser hecho dejando afuera un conjunto de datos y construir $T_1 \succ T_2 \succ \ldots \succ T_t$ con los otros datos. Entonces los casos en el conjunto excluido pueden ser clasificados usando cada árbol, dando una estimación del error verdadero de cada árbol. El α asociado al árbol que minimiza el error puede ser usado como parámetro de complejidad para podar el árbol construido con todo el entrenamiento.

Árboles de regresión

Como su nombre indica, CART también soporta la construcción de árboles de regresión. Árboles de regresión son algo más simples que los árboles de clasificación, porque los criterios de crecimiento y poda usados en CART son los mismos. La estructura es similar a un árbol de clasificación, excepto que cada hoja predice un número real. La resubstitución estimada es el error cuadrático medio,

$$R(S) = \frac{1}{n}\sum_i (y_i - h(c_i))^2 \tag{3.41}$$

donde y_i es el número real a predecir para el caso c_i y $h(c_i)$ es la predicción de y_i. El criterio de división es elegido para minimizar la estimación de resubstitución. La poda es hecha de una forma similar que la descrita anteriormente.

3.4. Regresión logística

3.4.1. Introducción

La regresión logística es muy conocida y ha sido bien estudiada, luego sólo se hará una breve introducción. Para mayor referencia ver [21] o [23]. La regresión logística es un tipo de regresión usada para predecir una variable binaria a partir de una o más variables predictoras o explicativas. Ésta intenta modelar la probabilidad de éxito usando una regresión lineal. La función logística es la base de este modelo,

$$f(z) = \frac{e^z}{e^z + 1} = \frac{1}{1 + e^{-z}} \tag{3.42}$$

donde z representa una combinación de variables explicativas y $f(z)$ representa la probabilidad de alguna salida particular, dado el conjunto de variables explicativas. La variable z es una medida de la contribución total de todas las variables independientes usadas en el modelo y es conocida como *logit*. Esta variable es usualmente definida como,

$$z = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_p x_p \tag{3.43}$$

donde β_0 es llamado el intercepto y $\beta_1, \ldots, \beta_p$ son llamados los coeficientes de regresión de $x_1, \ldots, x_p$, respectivamente.

3.4.2. Formulación

Al igual que antes, supongamos que tenemos N datos de entrenamiento. Cada dato es representado por un par (x_i, y_i), donde $x_i \in \mathbb{R}^p$, $y_i \in \{-1, 1\}$, $i = 1, \ldots, N$. x_i es un vector con las variables predictoras del caso i e y_i muestra a que clase pertenece. Las variables pueden ser continuas o discretas. Las variables discretas con dos o más categorías, son usualmente codificadas usando variables *dummy* (o variables indicadoras), que toman los valores 0 o 1. La salida y_i es descrita como teniendo distribución de Bernoulli, donde cada salida es determinada por una probabilidad no observada p_i. La idea básica de la regresión logística es usar el mecanismo ya desarrollado para regresión lineal, modelando la probabilidad p_i como una combinación lineal de las variables explicativas y un conjunto de coeficientes de regresión. La siguiente formulación expresa la regresión logística como un tipo de modelo lineal generalizado:

$$logit(p_i) = \log\left(\frac{p_i}{1-p_i}\right) = \beta_0 + \beta_1 (x_i)_1 + \ldots + \beta_p (x_i)_p \tag{3.44}$$

La probabilidad p_i y los coeficientes de regresión no son observados, éstos son típicamente determinados por algún tipo de optimización, por ejemplo, estimación por máxima verosimilitud, que encuentra valores que se ajustan mejor a los datos. Una fórmula equivalente a (3.44) usa la inversa de la función *logit*, la cual es la función logística:

$$p_i = logit^{-1}(\beta_0 + \beta_1 (x_i)_1 + \ldots + \beta_p (x_i)_p) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 (x_i)_1 + \ldots + \beta_p (x_i)_p)}} \tag{3.45}$$

Capítulo 4

Tratamiento de datos

4.1. Datos de fraude

Existen algunos problemas con las base del *Data WareHouse*, de fraude y la base réplica:

- Las transacciones fraudulentas no están identificadas en las bases del *Data WareHouse*.
- No existe una forma automática o directa de relacionar las transacciones entre la base de fraude y la base réplica, ya que no existe en ambas un código o ID de transacción. Tampoco se pueden relacionar estas bases con las del *Data WareHouse*, por falta de algún identificador común.
- No se tiene acceso al total de transacciones de la base réplica, luego no se podría hacer un modelo de detección (o clasificación), ya que sólo se tiene las transacciones fraudulentas y no las restantes en esta base.

Para solucionar estos problemas se manejaron dos opciones, marcar las transacciones fraudulentas en las tres bases transaccionales (mencionadas en 2.5) o mediante el sistema de crédito encontrar una manera de acceder a todas las transacciones ya con la marca de fraude. La primera opción era una tarea que requería un tiempo considerable, pues era una proceso prácticamente manual, luego se intentó la segunda opción. Pero finalmente no fue posible implementar esta opción por problemas técnicos. Durante el tiempo que se intentó, se construyó un modelo de detección de fraude a nivel cliente, en el cual se identificó si un cliente es propenso a sufrir fraude y está detallado en el apéndice B. Esto se hizo porque fue posible identificar a los clientes que sufrieron fraude, gracias a que el n° de cuenta sirvió de identificador común.

Por el tiempo disponible se decidió marcar seis meses de transacciones fraudulentas, de un periodo reciente. Las transacciones identificadas como fraudulentas se encuentran en la base fraude, pero esta base no tiene el tipo de transacción, es decir, si es venta en el comercio propio o no propio, o una transacción financiera, etc. Así a los datos en la base fraude se le añadió esta

Clasificación	N° de Trx's	Monto (M$)	% N° Trx's	% Monto
TF	643	292.096	14 %	67 %
Venta Propia	223	26.631	5 %	6 %
Venta No Propia	3.417	105.415	73 %	24 %
Exclusiones	437	8.414	9 %	2 %
Total	4.720	432.556	100 %	100 %

Tabla 4.1: Clasificación n° 1 de fraude, datos de un periodo de seis meses.

información mediante la base réplica, manualmente. Como la base fraude y las tres bases a nivel transaccional no comparten una variable identificadora, se buscó identificar las transacciones con la cuenta, fecha, monto y n° de cuotas, pero sólo se logró hacer un cruce de alrededor el 30 % del total de transacciones, el resto se marcó manualmente, transacción por transacción. Entre las transacciones en la base de fraude existen registros que se excluyeron:

- Gastos e impuestos asociados a TF: estos son registros que se encuentran en la base de fraude, pero no son transacciones propiamente tal, sino que están asociadas a una transacción financiera.
- Transacciones por montos <$2.000: se decidió no considerar transacciones por montos tan pequeños. Se escogió este monto pues existen varias transacciones fraudulentas de $2.000, $3.500 y $5.000, correspondientes a recargas telefónicas, que como se verá más adelante es un rubro importante para el fraude.
- Seguros: en los seis meses hay sólo ocho transacciones fraudulentas en el rubro de los seguros, pero éstos son un porcentaje importante del total de transacciones, luego la tasa de fraude es casi nula.

En la tabla 4.1 se muestra una primera clasificación de las transacciones de la base fraude. "Venta Propia" y "Venta No Propia" corresponden a las transacciones no financieras en el comercio propio y no propio, respectivamente y "TF" a las transacciones financieras. Con las exclusiones mencionadas arriba sólo se excluye el 9 % de las transacciones fraudulentas, lo que corresponde al 2 % del monto total de fraude.

Posteriormente se decidió que no se considerarían las transacciones no financieras hechas en el comercio propio, pues el n° de transacciones y monto de fraude son cercanos al 5 % de la base de fraude y estas transacciones son un gran porcentaje de la venta total, luego la tasa de fraude es muy baja. Así finalmente las transacciones que se consideran fraudulentas en este periodo se detallan en la tabla 4.2.

La "Venta No Propia" se obtiene de la base no propia y "TF" se obtiene de la base de transacciones financieras. En ambas bases se marcaron las transacciones fraudulentas y se filtraron todas las transacciones menores a $2.000 y las del rubro de los seguros. Estas bases se consolidaron en una sola, llamada en adelante "base consolidada", la cual será utilizada para construir los modelos.

Clasificación	N° de Trx's	Monto (M$)	% N° Trx's	% Monto
Venta No Propia	3.417	105.415	84 %	26 %
TF	643	292.096	16 %	74 %
Total	4.060	397.511	100 %	100 %

Tabla 4.2: Clasificación n° 2 de fraude, datos de un periodo de seis meses.

En la base consolidada se tiene una variable con el rubro, la cual tiene más de 200 valores, los cuales son estándares para la marca de la tarjeta de crédito (operador de la tarjeta). Esta variable se puede usar para clasificar las transacciones y da luz de la gran variabilidad de fraude, lo que hace difícil generar un solo modelo, además de las pocas transacciones fraudulentas que se tienen. En la tabla 4.3 se muestra una agrupación de esta variable en 19 categorías.

Sombreados se destacan los rubros a los que se aplicarán los modelos por la cantidad de transacciones y montos (Telefonía, Centros de pago y Trx. financieras). Con estos tres rubros (las Trx. financieras se unen en un rubro) se abarca el 57 % de transacciones y el 82 % del monto de fraude, luego se considera la gran mayoría del monto total de fraude. Para comparar estos modelos también se construirá un modelo con el total de las transacciones.

N	Clasificación	N° de Trx's	Monto (M$)	% N° de Trx's	% Monto
1	Viajes	41	5.147	1 %	1 %
2	Supermercados/Alimentos	115	5.248	3 %	1 %
3	Salud/Farmacias	83	3.826	2 %	1 %
4	Combustibles	112	5.096	3 %	1 %
5	Estaciones de servicio	291	8.463	7 %	2 %
6	Telefonía	1.093	14.747	27 %	4 %
7	Restaurant	40	973	1 %	0 %
8	Automotriz	4	357	0 %	0 %
9	Clubs/Bares	33	1.008	1 %	0 %
10	Apuestas	31	2.215	1 %	1 %
11	Centros de pago	577	19.508	14 %	5 %
12	Publicidad/Internet	263	5.289	6 %	1 %
13	Servicios profesionales	132	5.759	3 %	1 %
14	Entretenimiento	308	6.341	8 %	2 %
15	Tienda por departamentos	27	3.173	1 %	1 %
16	Artículos para el hogar	41	5.503	1 %	1 %
17	TF Bcos, transf	393	133.803	10 %	34 %
18	TF Propias	250	158.293	6 %	40 %
19	Otros Rubros	226	12.761	6 %	3 %
	Total	4.060	397.511	100 %	100 %

Tabla 4.3: Clasificación de fraude según rubro.

4.2. Variables relevantes

Las variables disponibles en la base consolidada se encuentran en la tabla 4.4.

Para identificar algún patrón en las transacciones fraudulentas se crearon variables acumuladoras, que contienen información histórica del cliente en la agrupación de rubro de la transacción entrante. Para hacer esto se particionó la base consolidada con la variable "ClassRubro" en 19 bases y en cada una de estas bases se calcularon las variables acumuladoras, que finalmente se agregaron a la base consolidada. El primer tipo de variable acumuladora corresponde a las transacciones hechas en el día en la misma agrupación de rubro. Esta variable se calculó para las transacciones que tienen hora, ya que con esta información es posible ordenar las transacciones. Para la base total las transacciones con hora son el 86 %. Esto se hizo con una rutina hecha en SPSS (*Statistical Package for the Social Sciences*), el cual es un programa estadístico informático muy usado en las ciencias sociales y las empresas de investigación de mercado. Las otras variables acumuladoras corresponden a las transacciones hechas en los últimos 30, 90 días y los últimos 12 meses. Las dos primeras dependen del día de la transacción, es decir, se utilizan las transacciones hechas en la misma agrupación de rubro en los últimos 30 y 90 días, desde el día de la transacción. La tercera depende del mes, se usan las transacciones hechas en la misma agrupación de rubro en los últimos 12 meses, sin contar el mes de la transacción. Esto se hizo así, pues las dos primeras son más sensibles al día de la transacción y se pretende capturar la historia reciente del cliente, en cambio la última pretende capturar un comportamiento histórico del cliente, algo más regular. Para hacer esto también se creó una rutina en SPSS. En la tabla 4.5 se encuentra una lista de las variables acumuladoras creadas.

Ahora que la base consolidada tiene todas las variables a considerar, variables de la transacción y acumuladoras, hay que elegir que variables entrarán a los modelos. Para esto se agruparon las variables en categorías acordes a cada variable, según el rango y la concentración de fraude. Luego se eligieron las variables analizando el poder discriminante de cada variable, calculando el KS (distancia de Kolmogorov-Smirnov, ver el apéndice A) de cada variable con respecto al "flag_fraude" y también se consideró la tasa de fraude en cada categoría. Se usa el KS, pues es un indicador conocido y usado en el negocio. Como se mencionó anteriormente se aplicarán modelos a la base total, Telefonía, Centros de Pago y Trx. financieras, así que las elecciones de variables se hacen en cada una de estas bases, lo cual es detallado en las siguientes secciones.

N°	Variable	N° Descripción
1	ID_Trx	Variable identificadora única
2	Cuenta	N° de cuenta del cliente
3	Fecha	Fecha de la transacción
4	Hora	Hora de la transacción
5	Tipo_Trx	Tipo de transacción
6	Rubro	Rubro
7	Comercio	Comercio
8	Monto	Monto en pesos
9	Cuotas	N° de cuotas
10	ClassRubro	Agrupación del Rubro en 19 categorías
11	flag_fraude	Flag si la transacción es fraude

Tabla 4.4: Variables de la base consolidada.

N°	Variable	N° Descripción
1	N_dia	N° de transacciones previas del día
2	Monto_dia	Monto acumulado del día
3	MontoProm_dia	Monto promedio de las transacciones previas del día
4	N_30d	N° de transacciones de los últimos 30 días
5	Monto_30d	Monto acumulado en los últimos 30 días
6	MontoProm_30d	Monto promedio en los últimos 30 días
7	MontoMax_30d	Monto máximo en los últimos 30 días
8	RelMontoProm_30d	Relación entre el monto y el monto promedio de 30 días
9	RelMontoMax_30d	Relación entre el monto y el monto máximo de 30 días
10	N_90d	N° de transacciones de los últimos 90 días
11	Monto_90d	Monto acumulado en los últimos 90 días
12	MontoProm_90d	Monto promedio en los últimos 90 días
13	MontoMax_90d	Monto máximo en los últimos 90 días
14	RelMontoProm_90d	Relación entre el monto y el monto promedio de 90 días
15	RelMontoMax_90d	Relación entre el monto y el monto máximo de 90 días
16	N_12M	N° de transacciones de los últimos 12 meses
17	Monto_12Md	Monto acumulado en los últimos 12 meses
18	MontoProm_12M	Monto promedio en los últimos 12 meses
19	MontoMax_12M	Monto máximo en los últimos 12 meses
20	RelMontoProm_12M	Relación entre el monto y el monto promedio de 12 meses
21	RelMontoMax_12M	Relación entre el monto y el monto máximo de 12 meses

Tabla 4.5: Variables acumuladoras.

4.2.1. Base total

En la tabla 4.6 se muestra el KS de las variables a considerar en los modelos para la base total.

Las variables “Monto_Ag”, “Cuotas_Ag” y “ClassRubro_Ag” entrarán en los modelos, pues aunque las dos primeras tienen bajo KS, estas variables son propias de la transacción e importantes para el negocio, la última entra también por ser la de más alto KS. De las variables acumuladoras se puede ver que las con más alto KS son las del día y las de 12 meses, pero no se elegirán solo de estas variables, ya que están muy correlacionadas, por ejemplo existe una alta correlación entre “RelMontoProm_12M_Ag” y “RelMontoMax_12M_Ag”, y lo mismo ocurre cuando se analizan variables del mismo periodo (día, 30 días, etc.). En la tabla 4.7 se encuentran las correlaciones de Pearson de las variables acumuladoras de 90 días, ésta fue calculada en una base de tres millones de registros aproximadamente.

Tres pares de variables tienen correlación mayor a 0,6. Aunque dentro del mismo periodo no todas las variables tienen correlación alta, para reducir los grados de libertad de los modelos se decidió elegir sólo una variable por periodo. Finalmente las variables que entrarán en los modelos serán siete variables: “Monto_Ag”, “Cuotas_Ag”, “ClassRubro_Ag”, “Monto_dia_Ag”, “Monto_30d_Ag”, “MontoProm_90d_Ag” y “RelMontoMax_12M_Ag”. En las tablas 4.8 - 4.14 se encuentran las categorías de estas variables y el cálculo del KS.

N°	Variable	KS
1	Monto_Ag	9%
2	Cuotas_Ag	18%
3	ClassRubro_Ag	56%
4	Monto_dia_Ag	42%
5	MontoProm_dia_Ag	42%
6	Monto_30d_Ag	19%
7	MontoProm_30d_Ag	17%
8	RelMontoProm_30d_Ag	15%
9	RelMontoMax_30d_Ag	15%
10	Monto_90d_Ag	33%
11	MontoProm_90d_Ag	36%
12	RelMontoProm_90d_Ag	33%
13	RelMontoMax_90d_Ag	33%
14	Monto_12M_Ag	45%
15	RelMontoProm_12M_Ag	45%
16	RelMontoMax_12M_Ag	47%

Tabla 4.6: KS de variables para la base total.

	Monto_Ag	MontoProm_Ag	RelMontoProm_Ag	RelMontoMax_Ag
Monto_Ag	1	0,699	0,386	0,178
MontoProm_Ag	0,699	1	0,700	0,614
RelMontoProm_Ag	0,386	0,700	1	0,906
RelMontoMax_Ag	0,178	0,614	0,906	1

Tabla 4.7: Correlación de las variables acumuladoras de 90 días.

Monto_Ag	% Normal	% Fraude	KS
$\leq$ 5.000	22%	13%	9%
5.001 - 10.000	18%	21%	-2%
10.001 - 20.000	33%	34%	-1%
20.001 - 50.000	11%	10%	2%
50.001 - 100.000	8%	7%	1%
100.001 - 200.000	4%	7%	-4%
200.001 - 500.000	2%	3%	-2%
$\geq$ 500.001	1%	5%	-4%
Total	100%	100%	9%

Tabla 4.8: Variable Monto_Ag para el modelo de la base total.

Cuotas_Ag	% Normal	% Fraude	KS
0	20 %	30 %	-11 %
1	37 %	19 %	17 %
2 -6	40 %	39 %	1 %
7 - 12	3 %	6 %	-3 %
$\geq$ 13	1 %	5 %	-4 %
Total	100 %	100 %	18 %

Tabla 4.9: Variable Cuotas_Ag para el modelo de la base total.

ClassRubro_Ag	% Normal	% Fraude	KS
Tasa fraude menor a la mitad	69 %	13 %	56 %
Tasa fraude entre la mitad y el doble	14 %	13 %	0 %
Tasa fraude entre el doble y 4 veces	8 %	25 %	-17 %
Tasa fraude 4 veces o más	9 %	48 %	-39 %
Total	100 %	100 %	56 %

Tabla 4.10: Variable ClassRubro_Ag para el modelo de la base total.

Monto_dia_Ag	% Normal	% Fraude	KS
S/Hora	14 %	24 %	-10 %
0	79 %	38 %	42 %
1 - 50.000	5 %	21 %	-16 %
$\geq$ 50.001	2 %	18 %	-16 %
Total	100 %	100 %	42 %

Tabla 4.11: Variable Monto_dia_Ag para el modelo de la base total.

Monto_30d_Ag	% Normal	% Fraude	KS
0	41 %	56 %	-15 %
1 - 50.000	34 %	18 %	16 %
50.001 - 500.000	22 %	20 %	3 %
$\geq$ 500.001	2 %	6 %	-3 %
Total	100 %	100 %	19 %

Tabla 4.12: Variable Monto_30d_Ag para el modelo de la base total.

MontoProm_90d_Ag	% Normal	% Fraude	KS
0	22 %	55 %	-33 %
1 - 200.000	77 %	41 %	36 %
$\geq$ 200.001	1 %	4 %	-3 %
Total	100 %	100 %	36 %

Tabla 4.13: Variable MontoProm_90d_Ag para el modelo de la base total.

RelMontoMax_12M_Ag	% Normal	% Fraude	KS
0	14 %	58 %	-45 %
0,01 - 1,50	80 %	33 %	47 %
$\geq$ 1,51	6 %	9 %	-3 %
Total	100 %	100 %	47 %

Tabla 4.14: Variable RelMontoMax_12M_Ag para el modelo de la base total.

4.2.2. Base telefonía

En el rubro de telefonía se usarán sólo las transacciones de recargas telefónicas, pues abarcan casi la totalidad del fraude en este rubro. Así la base será llamada base de recargas telefónicas. En la tabla 4.15 se muestra el KS de las variables a considerar en los modelos para la base de recargas telefónicas.

Al igual que en la base total, las variables "MarcaMonto", "Cuotas_Ag" y "Comercio", propias de la transacción entrarán en los modelos, aunque la primera entra también por ser la de más alto KS. De las variables acumuladoras se puede ver que las con más alto KS son la de 12 meses, pero no se elegirán sólo estas variables, por la misma razón que antes, tienen alta correlación y para reducir los grados de libertad. En este caso se tienen correlaciones altas como en la tabla 4.7. Finalmente las variables que entrarán en los modelos son siete variables: "MarcaMonto", "Cuotas_Ag", "Comercio", "Monto_dia_Ag", "MontoProm_30d_Ag", "RelMontoProm_90d_Ag" y "RelMontoMax_12M_Ag". En las tablas 4.16 - 4.22 se encuentran las categorías de estas variables y el cálculo del KS.

N°	Variable	KS
1	MarcaMonto	74 %
2	Cuotas_Ag	21 %
3	Comercio	27 %
4	N_dia_Ag	54 %
5	Monto_dia_Ag	55 %
6	Monto_30d_Ag	49 %
7	MontoProm_30d_Ag	51 %
8	RelMontoProm_30d_Ag	50 %
9	RelMontoMax_30d_Ag	47 %
10	Monto_90d_Ag	61 %
11	MontoProm_90d_Ag	65 %
12	RelMontoProm_90d_Ag	67 %
13	RelMontoMax_90d_Ag	67 %
14	Monto_12M_Ag	66 %
15	MontoProm_12M_Ag	67 %
16	RelMontoProm_12M_Ag	74 %
17	RelMontoMax_12M_Ag	74 %

Tabla 4.15: KS de variables para la base de recarga telefónicas.

MarcaMonto	% Normal	% Fraude	KS
$<$ 10.000	93 %	19 %	74 %
$\geq$ 10.000	7 %	81 %	-74 %
Total	100 %	100 %	74 %

Tabla 4.16: Variable MarcaMonto para el modelo de la base total.

Cuotas_Ag	% Normal	% Fraude	KS
1	56 %	57 %	0 %
2 - 4	36 %	16 %	21 %
5	1 %	2 %	-1 %
6	6 %	25 %	-20 %
Total	100 %	100 %	21 %

Tabla 4.17: Variable Cuotas_Ag para el modelo de la base recargas telefónicas.

Comercio	% Normal	% Fraude	KS
Comercio 1	50 %	22 %	27 %
Comercio 2	32 %	40 %	-8 %
Comercio 3	18 %	38 %	-20 %
Total	100 %	100 %	27 %

Tabla 4.18: Variable Comercio para el modelo de la base recargas telefónicas.

Monto_dia_Ag	% Normal	% Fraude	KS
S/Hora, 0	95 %	42 %	54 %
1 - 4.999	3 %	2 %	1 %
$\geq$ 5.000	2 %	57 %	-55 %
Total	100 %	100 %	55 %

Tabla 4.19: Variable Monto_dia_Ag para el modelo de la base recargas telefónicas.

MontoProm_30d_Ag	% Normal	% Fraude	KS
0	30 %	76 %	-46 %
1 - 9.999	68 %	16 %	51 %
$\geq$ 10.000	3 %	8 %	-5 %
Total	100 %	100 %	51 %

Tabla 4.20: Variable MontoProm_30d_Ag para el modelo de la base recargas telefónicas.

RelMontoProm_90d_Ag	% Normal	% Fraude	KS
0	12 %	73 %	-61 %
0,01 - 2,00	86 %	19 %	67 %
$\geq$ 2,01	2 %	8 %	-6 %
Total	100 %	100 %	67 %

Tabla 4.21: Variable RelMontoProm_90d_Ag para el modelo de la base recargas telefónicas.

RelMontoMax_12M_Ag	% Normal	% Fraude	KS
0	10 %	76 %	-66 %
0,01 - 1,75	89 %	15 %	74 %
$\geq$ 1,76	2 %	9 %	-8 %
Total	100 %	100 %	74 %

Tabla 4.22: Variable RelMontoMax_12M_Ag para el modelo de la base recargas telefónicas.

4.2.3. Base transacciones financieras

En la tabla 4.23 se muestra el KS de las variables a considerar en los modelos para la base de transacciones financieras.

Siguiendo la misma lógica que antes las variables que entrarán en los modelos son las siete siguientes: “Monto_Ag”, “Cuotas_Ag”, “ClassRubro_Ag”, “PromMonto_dia_Ag”, “Monto_30d_Ag”, “Monto_90d_Ag” y “RelMontoMax_12M_Ag”. En las tablas 4.24 - 4.30 se encuentran las categorías de estas variables y el cálculo del KS.

N°	Variable	KS
1	Monto_Ag	24 %
2	Cuotas_Ag	18 %
3	ClassRubro_Ag	34 %
4	Monto_dia_Ag	33 %
5	MontoProm_dia_Ag	33 %
6	Monto_30d_Ag	12 %
7	MontoProm_30d_Ag	7 %
8	RelMontoProm_30d_Ag	6 %
9	RelMontoMax_30d_Ag	7 %
10	Monto_90d_Ag	15 %
11	MontoProm_90d_Ag	12 %
12	RelMontoProm_90d_Ag	11 %
13	RelMontoMax_90d_Ag	11 %
14	Monto_12M_Ag	38 %
15	MontoProm_12M_Ag	39 %
16	RelMontoProm_12M_Ag	38 %
17	RelMontoMax_12M_Ag	39 %

Tabla 4.23: KS de variables para la base de transacciones financieras.

Monto_Ag	% Normal	% Fraude	KS
$\leq$ 99.999	49 %	25 %	24 %
100.000 - 299.999	32 %	35 %	-4 %
$\geq$ 300.000	20 %	40 %	-20 %
Total	100 %	100 %	24 %

Tabla 4.24: Variable Monto_Ag para el modelo de la base trx. financieras.

Cuotas_Ag	% Normal	% Fraude	KS
1	3 %	4 %	-1 %
2 - 6	50 %	31 %	19 %
7 - 12	30 %	35 %	-5 %
13 - 24	8 %	19 %	-11 %
25 - 36	8 %	8 %	0 %
$\geq$ 37	1 %	4 %	-3 %
Total	100 %	100 %	18 %

Tabla 4.25: Variable Cuotas_Ag para el modelo de la base trx. financieras.

ClassRubro_Ag	% Normal	% Fraude	KS
TF propias	72 %	39 %	34 %
TF no propias	28 %	61 %	-34 %
Total	100 %	100 %	34 %

Tabla 4.26: Variable ClassRubro_Ag para el modelo de la base trx. financieras.

MontoProm_dia_Ag	% Normal	% Fraude	KS
0	94 %	62 %	33 %
1 - 100.000	2 %	6 %	-4 %
$\geq$ 100.001 o S/Hora	3 %	32 %	-29 %
Total	100 %	100 %	55 %

Tabla 4.27: Variable MontoProm_dia_Ag para el modelo de la base trx. financieras.

Monto_30d_Ag	% Normal	% Fraude	KS
0	77 %	71 %	6 %
1 - 100.000	14 %	8 %	6 %
100.001 - 200.000	4 %	5 %	0 %
$\geq$ 200.001	5 %	16 %	-11 %
Total	100 %	100 %	12 %

Tabla 4.28: Variable Monto_30d_Ag para el modelo de la base trx. financieras.

Monto_90d_Ag	% Normal	% Fraude	KS
0	54 %	66 %	-11 %
1 - 200.000	29 %	14 %	15 %
$\geq$ 200.001	17 %	20 %	-4 %
Total	100 %	100 %	15 %

Tabla 4.29: Variable Monto_90d_Ag para el modelo de la base trx. financieras.

RelMontoMax_12M_Ag	% Normal	% Fraude	KS
0	27 %	65 %	-38 %
0,01 - 2,00	63 %	26 %	37 %
2,01 - 3,50	4 %	2 %	2 %
$\geq$ 3,51	6 %	7 %	-1 %
Total	100 %	100 %	39 %

Tabla 4.30: Variable RelMontoMax_12M_Ag para el modelo de la base trx. financieras.

4.2.4. Base centros de pago

En la tabla 4.31 se muestra el KS de las variables a considerar en los modelos para la base centros de pago.

Llama la atención que una variable propia de la transacción como el n° de cuotas tenga un KS tan alto (79 %), siendo el segundo más alto de las variables en 4.31. La razón por la cual esta variable tiene este poder discriminante, es porque el 79 % de las transacciones fraudulentas tiene tres cuotas, pero sólo un 6 % de las transacciones tiene este n° de cuotas, como se puede ver en la tabla 4.32.

Esta concentración de transacciones fraudulentas es extraña, por lo que se aislaron las transacciones con tres cuotas. Al analizar los comercios se encontró que el 95 % de las transacciones fraudulentas con tres cuotas pertenecían a un solo comercio, el cual se identificará como "comercio F". En la tabla 4.33 se muestra la variable "Cuotas" en el "comercio F". El 80 % de las transacciones fraudulentas de la base centros de pago se encuentran en este comercio.

En el "comercio F" se tiene una tasa de fraude del 4,4 % y en las transacciones con tres cuotas se tiene una del 25 %, es decir, una de cada cuatro transacciones es fraudulenta, lo cual es demasiado alto. Si se supone un porcentaje de ganancia del 10 % sobre la venta, en este tipo de transacciones se estaría ganando aproximadamente seis millones de pesos, pero se estarían perdiendo doce millones, luego no genera ganancia dejar pasar éstas transacciones. Por esta razón y porque este porcentaje de transacciones fraudulentas es atípico para un sólo comercio, no se aplicará ningún modelo a las transacciones del rubro "Centros de pago", donde el 80 % del fraude corresponde al "comercio F". Otra razón para no aplicar modelos en este rubro es que los resultados que se tendrían no serían comparables con los otros rubros.

N°	Variable	KS
1	Monto_Ag	41 %
2	Cuotas_Ag	79 %
3	Monto_dia_Ag	77 %
4	MontoProm_dia_Ag	77 %
5	Monto_30d_Ag	59 %
6	MontoProm_30d_Ag	40 %
7	RelMontoProm_30d_Ag	25 %
8	RelMontoMax_30d_Ag	49 %
9	Monto_90d_Ag	69 %
10	MontoProm_90d_Ag	59 %
11	RelMontoProm_90d_Ag	52 %
12	RelMontoMax_90d_Ag	58 %
13	Monto_12M_Ag	71 %
14	MontoProm_12M_Ag	80 %
15	RelMontoProm_12M_Ag	76 %
16	RelMontoMax_12M_Ag	55 %

Tabla 4.31: KS de variables para la base centros de pago.

Cuotas_Ag	% Normal	% Fraude	KS
0	8 %	15 %	-6 %
1	84 %	5 %	78 %
2	0 %	1 %	0 %
3	6 %	79 %	-72 %
4+	1 %	1 %	0 %
Total	100 %	100 %	79 %

Tabla 4.32: Variable Cuotas_Ag para el modelo de la base centros de pago.

Cuotas	Trx. Normal	Trx. Fraude	% Normal	% Fraude	KS
0	34	3	0 %	1 %	-0 %
1	8.623	27	83 %	6 %	77 %
2	471	0	5 %	0 %	5 %
3	1.269	429	12 %	93 %	-81 %
Total	10.397	459	100 %	100 %	82 %

Tabla 4.33: Variable Cuotas en el comercio F.

4.2.5. Comparación de variables entre rubros

Para las recargas telefónicas se tienen varias variables con KS muy altos, como se pueden ver en la tabla 4.15. Existen dos variables con KS más bajo (21 % y 27 %), las demás son en general mayor al 50 %, existiendo algunas con el 74 % de discriminación. Esto ocurre pues en las transacciones fraudulentas de esta base se encuentra un claro patrón de fraude: el monto máximo diario de recargas es $60.000 y por transacción es $25.000, así se repite el fraude de seis transacciones seguidas de $10.000 o dos transacciones de $25.000 y una de $10.000.

En las transacciones financieras no se tiene ninguna variable con un KS mayor al 40 % y se tienen tres variables con un KS menor al 7 %. Acá no se encuentra ningún patrón claro, las transacciones fraudulentas se confunden con las normales.

Por último en la base total se tienen valores de KS entre el 12 % y 56 %.

En general en la base de recargas se tienen variables con un mayor poder discriminante que las otras bases y en la base total se tienen variables con mayor poder discriminante que la base de TF.

Capítulo 5

Entrenamiento y test

Con las bases listas y las variables seleccionadas, se pueden generar las bases para construir los modelos y las bases para evaluarlos. Las primeras son las bases de entrenamiento y las segundas las bases de test.

5.1. Bases de entrenamiento

Como las transacciones fraudulentas disponibles son escazas, se usará un 80 % para entrenamiento y 20 % para test, porcentajes estándares en la generación de modelos. Se utilizarán tres tipos de bases de entrenamiento:

1. Una base equilibrada, con 50 % de transacciones fraudulentas y 50 % transacciones normales, este tipo de bases son usadas en el negocio. Denotada E50.
2. Una base con 25 % de transacciones fraudulentas y 75 % transacciones normales. Denotada E25.
3. Una base con 10 % de transacciones fraudulentas y 90 % transacciones normales. Denotada E10.

Las dos últimas bases son propuestas en este trabajo, ya que las transacciones fraudulentas son un pequeño porcentaje del total de transacciones y se trata de capturar este fenómeno. No se probaron porcentajes menores al 10 % de transacciones fraudulentas, pues la base de entrenamiento crecía demasiado de tamaño y dada la capacidad de cálculo disponible era difícil hacer el entrenamiento (algunos tiempos de entrenamiento de la sección 5.2 son mayores a un día). En la tabla 5.1 se tienen las cantidades de transacciones normales y fraudulentas en las bases de entrenamiento de las bases total, recargas y TF.

Las transacciones usadas para el entrenamiento no entran posteriormente en las bases de test, para hacer la evaluación de los modelos independiente de la construcción de éstos.

Base	Base entrenamiento	N° trx fraude	N° trx normal	N° total
Total	E50	3.260	3.260	6.520
	E25		9.780	13.040
	E10		29.340	32.600
Recargas	E50	838	838	1.676
	E25		2.514	3.352
	E10		7.542	8.380
TF	E50	514	514	1.028
	E25		1.542	2.056
	E10		4.626	5.140

Tabla 5.1: Cantidad de transacciones normales y fraudulentas en las bases de entrenamiento.

5.2. Entrenamiento modelos

Para el entrenamiento de los modelos se usará el programa R. R es un lenguaje y entorno de programación para análisis estadístico y gráfico. Se trata de un proyecto de software libre y se distribuye bajo la licencia GNU GPL (licencia pública general de GNU). Además de ser de libre uso, es gratuito. En R es posible bajar paquetes con programas hechos por otros usuarios. Usando esta modalidad se utilizarán cuatro paquetes correspondientes a los modelos que se usarán en R:

- "e1071": este paquete contiene una función que implementa SVM, específicamente usa la librería *libsvm*.
- "nnet": este paquete contiene una función del mismo nombre que implementa redes neuronales *feed-forward* con una capa oculta.
- "rpart": este paquete contiene una función del mismo nombre que implementa CART.
- "RWeka": este paquete contiene una función que implementa C4.5, específicamente usa J48, una implementación de código abierto hecha en Java.

Para el modelo faltante, regresión logística, se usó el programa SPSS. La razón de usar SPSS para la regresión logística es que en el negocio este modelo es conocido y se ha puesto en práctica usando este programa.

5.2.1. Entrenamiento

Para hacer el entrenamiento se usará la función *tune* de R. Esta función sirve para ajustar los parámetros del modelo a los datos de entrenamiento. Se le entregan rangos o vectores con los parámetros a ajustar y hace una búsqueda exhaustiva o de fuerza bruta sobre estos rangos. El modelo elegido es el que tiene mejor rendimiento, el cual se mide con el error de clasificación.

Además se incluye la opción de usar validación cruzada. La validación cruzada es una técnica utilizada para evaluar los resultados de un análisis estadístico y garantizar que son independientes de la partición entre datos de entrenamiento o prueba. Esta consiste en repetir y calcular la media aritmética obtenida de las medidas de evaluación sobre diferentes particiones. El tipo de validación cruzada que se usará será validación cruzada de diez iteraciones (*10-fold cross-validation*), en la cual los datos se dividen en diez subconjuntos, uno de los subconjuntos se utiliza como datos test y el resto (nueve) como datos de entrenamiento. El proceso se repite diez veces con cada uno de los posibles datos de prueba, finalmente se realiza la media aritmética de los resultados de cada repetición para obtener un único resultado.

Los modelos en R que aceptan la función *tune* son SVM, CART y redes neuronales. Para el modelo C4.5 se creó una rutina que calcula el error de clasificación mediante validación cruzada y elige el set de parámetros con mejor rendimiento. Como en la regresión logística no hay que elegir parámetros y ha sido plenamente estudiada, no se hizo validación cruzada.

A continuación se listan los parámetros y rangos usados para ajustar los modelos:

- SVM polinomial
  - Grado (d): es el grado del polinomio del *kernel* o núcleo. Los valores usados fueron 2, 3 y 4.
  - Gamma (γ): es la constante del polinomio. Los valores usados fueron $0,1$ y 1.
  - Costo (C): es el costo de la violación de restricciones (el término γ del lagrangiano 3.19 visto en la sección 3.2.2). Los valores usados fueron 1 y 10.
  - N° de modelos: 12.
- SVM radial
  - Gamma (γ): parámetro de ajuste del *kernel* radial. Los valores usados fueron $0,01$; $0,1$; 1 y 10.
  - Costo (C): es el costo de la violación de restricciones (el término γ del lagrangiano 3.19 visto en la sección 3.2.2). Los valores usados fueron $0,1$; 1 y 10.
  - N° de modelos: 12.
- CART
  - Minsplit (*minS*): es el número mínimo de casos en un nodo para que haya una división. Los valores usados fueron 2, 5, 10 y 20.
  - Complejidad (cp): para que una división sea hecha, ésta debe mejorar el ajuste (medido con la ganancia de información) por al menos este factor cp. Los valores usados fueron $0,01$ y $0,001$.
  - N° de modelos: 8.
- C4.5

Modelos	E50	E25	E10
SVM_rad	2.808	10.052	46.872
SVM_pol	9.870	49.986	172.445[2]
CART	39	101	545
C4.5	12	16	43
ANN	14.848	28.769	48.437[2]

Tabla 5.2: Tiempos de entrenamiento en segundos, base total.

- MinHojas (*minH*): número mínimo de casos por hoja. Los valores usados fueron 2, 5, 10 y 20.
- NCapas (*N*): número de iteraciones para reducir el error de poda (técnica similar a la validación cruzada). Los valores usados fueron 3, 5 y 10.
- N° modelos: 12 modelos.

- ANN
  - Tamaño (*size*): número de nodos en la capa oculta. El rango de valores usados fueron los del intervalo $[5,20]$.
  - Decaimiento (*decay*): es un parámetro de regularización o *penalty* para evitar el *overfitting* [44, pág. 245]. Los valores usados fueron $0,1$; $0,01$ y $0,001$.
  - N° modelos: 48 modelos.

Los *kernels* que usa R son un poco diferentes de los vistos en 3.2.2, por lo que se muestran a continuación:

$$\begin{aligned} \textit{Polinomio de grado d} &: K(x,x') = (\gamma < x,x' >)^d \\ \textit{Base radial} &: K(x,x') = exp(-\gamma||x-x'||^2) \end{aligned}$$

Base total

En las tablas 5.2, 5.3 y 5.4 se muestran los tiempos de entrenamiento[2], los parámetros escogidos para cada modelo y el error de clasificación de los modelos para las tres bases de entrenamiento.

[2]Los tiempos de entrenamiento fueron medidos en un sistema operativo de 64 bits, con un procesador Pentium(R) Dual-Core CPU T4400 @ 2.20GHz $\times$ 2 y 4GB de memoria RAM. Como el tiempo estimado para los modelos SVM polinomial y ANN en la base E10 era entre dos y tres días, se usó otro computador con mayor poder de cálculo: sistema operativo de 32 bits, procesador Intel(R) Core(TM) i5 CPU 650 @ 3.20GHz 3.19 GHz y 3GB de RAM

	E50	E25	E10
SVM_rad	γ=0,1 C=10	γ=0,1 C=10	γ=0,1 C=10
SVM_pol	d=2 γ=1 C=1	d=4 γ=0,1 C=10	d=3 γ=0,1 C=10
CART	$minS$=2 cp=0,001	$minS$=2 cp=0,001	$minS$=2 cp=0,001
C4.5	$minH$=5 N=3	$minH$=10 N=3	$minH$=5 N=5
ANN	$size$=6 $decay$=0,1	$size$=7 $decay$=0,1	$size$=11 $decay$=0,1

Tabla 5.3: Parámetros escogidos para cada modelo según su rendimiento, base total.

Modelos	E50	E25	E10
SVM_rad	12.62	8.74	5.11
SVM_pol	12.51	8.86	5.20
CART	12.87	8.63	5.09
C4.5	12.73	9.18	5.07
ANN	12.56	8.80	5.02

Tabla 5.4: Errores de clasificación (%) en el entrenamiento, base total.

Los tiempos de entrenamiento fueron muy pequeños para C4.5, todos menor a un minuto. Para CART también fueron pequeños, todos menor a diez minutos. Para los otros tres modelos, los tiempos de entrenamiento fueron de varias horas, incluso días. Los cinco modelos siguen la tendencia lógica de necesitar más tiempo al aumentar el tamaño de la muestra de entrenamiento. El entrenamiento que tomó más tiempo fue para el modelo SVM polinomial, el cual demoró dos días, se estimaba una duración de tres días, por lo que se usó un computador con mayor poder de cálculo[2]. Dependiendo el modelo se probó un n° distinto de combinaciones de parámetros. En promedio el modelo que más demoró fue el SVM polinomial para la base E10, con un tiempo de 4 horas por modelo.

En cuanto a los parámetros, para los modelos SVM radial y CART, éstos fueron los mismos independiente de la muestra usada. Para el modelo ANN sólo cambió el n° de nodos en la capa oculta. Los dos modelos restantes mostraron mayor cambio de los parámetros dependiendo de la muestra.

Los errores de clasificación fueron calculados por validación cruzada. Todos los modelos siguieron la tendencia de disminuir el error a medida que se usan más casos de transacciones normales. Para la muestra E50 el menor error de clasificación se obtuvo con el modelo SVM polinomial, para E25 con el modelo CART y para E10 con ANN.

Base recargas

En las tablas 5.5, 5.6 y 5.7 se muestran los tiempos de entrenamiento, los parámetros escogidos para cada modelo y el error de clasificación de este modelo para las tres bases de entrenamiento.

Los tiempos de entrenamiento fueron muy pequeños para C4.5 y CART, todos menor a 61 segundos. Para los modelos de SVM los tiempos de entrenamiento fueron menores a 30 minutos y los tiempos para ANN fueron superiores a los restantes, siendo el mayor de más de tres horas. Los cinco modelos siguen la tendencia lógica de necesitar más tiempo al aumentar el tamaño de la muestra de entrenamiento. El entrenamiento que tomó más tiempo fue para el modelo ANN. En promedio el modelo que más demoró fue el ANN para la base E10, con un tiempo de 279 segundos por modelo.

Los únicos parámetros que permanecieron constantes según la muestra de entrenamiento

Modelos	E50	E25	E10
SVM_rad	117	301	1.017
SVM_pol	77	326	1.496
CART	9	23	61
C4.5	8	6	13
ANN	2488	5.279	13.376

Tabla 5.5: Tiempos de entrenamiento en segundos, base recargas.

Modelos	E50	E25	E10
SVM_rad	γ=1 C=10	γ=0,1 C=10	γ=1 C=1
SVM_pol	d=2 γ=0,1 C=10	d=3 γ=0,1 C=10	d=2 γ=0,1 C=10
CART	*minS*=2 cp=0,001	*minS*=10 cp=0,001	*minS*=10 cp=0,001
C4.5	*minH*=5 N=3	*minH*=2 N=5	*minH*=10 N=10
ANN	*size*=5 *decay*=0,1	*size*=5 *decay*=0,1	*size*=13 *decay*=0,1

Tabla 5.6: Parámetros escogidos para cada modelo según su rendimiento, base recargas.

Modelos	E50	E25	E10
SVM_rad	8.77	5.10	2.99
SVM_pol	8.83	4.95	3.08
CART	8.83	5.40	3.05
C4.5	8.71	5.16	3.07
ANN	8.59	5.04	2.86

Tabla 5.7: Errores de clasificación (%) en el entrenamiento, base recargas.

fueron C para el modelo SVM polinomial, cp para CART y $decay$ para ANN. Los demás parámetros variaron según la muestra de entrenamiento.

Todos los modelos siguieron la tendencia de disminuir el error de clasificación a medida que se usan más casos de transacciones normales. Para la muestra E50 el menor error de clasificación se obtuvo con el modelo ANN, para E25 con el modelo SVM polinomial y para E10 con ANN.

Base trx financieras

En las tablas 5.8, 5.9 y 5.10 se muestran los tiempos de entrenamiento, los parámetros escogidos para cada modelo y el error de clasificación de este modelo para las tres bases de entrenamiento.

Los tiempos de entrenamiento fueron muy pequeños para C4.5 y CART, todos menor a 35 segundos. Para los modelos de SVM los tiempos de entrenamiento fueron menores a 20 minutos y los tiempos para ANN son muy superiores a los restantes, siendo el mayor de casi tres horas. Los cinco modelos siguen la tendencia lógica de necesitar más tiempo al aumentar el tamaño de la muestra de entrenamiento. El entrenamiento que tomó más tiempo fue para el modelo ANN. En promedio el modelo que más demoró fue el ANN para la base E10, con un tiempo de 218 segundos por modelo.

En cuanto a los parámetros, para el modelo SVM radial, éstos fueron los mismos independiente de la muestra usada. Para los otros modelos, los parámetros que permanecieron constantes según la muestra de entrenamiento fueron cp para CART y $decay$ para ANN. Los demás parámetros variaron según la muestra de entrenamiento.

Todos los modelos siguieron la tendencia de disminuir el error de clasificación a medida que se usan más casos de transacciones normales. Para la muestra E50 el menor error de clasificación se obtuvo con el modelo ANN, para E25 con el modelo SVM radial y para E10 hubo un empate entre los modelos SVM.

Modelos	E50	E25	E10
SVM_rad	89	235	738
SVM_pol	77	243	939
CART	9	17	35
C4.5	8	5	9
ANN	2.007	4.149	10.474

Tabla 5.8: Tiempos de entrenamiento en segundos, base trx financieras.

Modelos	E50	E25	E10
SVM_rad	γ=0,1 C=10	γ=0,1 C=10	γ=0,1 C=10
SVM_pol	d=2 γ=1 C=1	d=2 γ=0,1 C=10	d=3 γ=0,1 C=10
CART	$minS$=10 cp=0,001	$minS$=2 cp=0,001	$minS$=20 cp=0,001
C4.5	$minH$=2 N=3	$minH$=10 N=3	$minH$=2 N=3
ANN	$size$=9 $decay$=0,1	$size$=8 $decay$=0,1	$size$=10 $decay$=0,1

Tabla 5.9: Parámetros escogidos para cada modelo según su rendimiento, base trx financieras.

Modelos	E50	E25	E10
SVM_rad	20.33	15.37	7.88
SVM_pol	20.34	15.47	7.88
CART	20.34	16.88	8.02
C4.5	20.53	16.00	8.21
ANN	20.23	16.00	8.13

Tabla 5.10: Errores de clasificación (%) en el entrenamiento, base trx financieras.

5.3. Test

Para las bases test se usaron dos tipos. Sean α, β y γ las tasas de fraude de las bases total, recargas telefónicas y transacciones financieras, respectivamente. Las bases test construidas fueron:

1. Una base que tuviera la tasa real de fraude, es decir, α, β o γ, según corresponda. Denotada Test*X*, donde *X* es la tasa real de fraude.

2. Una base con un 0,5 % como tasa de fraude. Denotada Test05.

Estos dos tipos de bases test se hicieron para ver si existe alguna diferencia en la predicción de los modelos cuando se cambia la proporción de transacciones fraudulentas.

En las tablas 5.11, 5.12 y 5.13 se pueden observar los tiempos de test en las muestras test con la tasa de fraude real para las bases total, recargas telefónicas y trx. financieras, respectivamente. Además se muestran los tiempos de entrenamiento promedio por modelo para compararlos.

Sólo se hacen comparaciones relativas entre los tiempos de entrenamiento y test, pues son procesos distintos. Primero los tiempos para la base total. Los tiempos de test para los modelos SVM fueron un orden de magnitud superiores a los restantes, siendo mayores para SVM radial. Los tiempos de test se mantuvieron prácticamente constantes para CART, C4.5 y ANN entrenados en las diferentes muestras, todos menores a 74 segundos. Aunque el tiempo de entrenamiento para SVM polinomial fue superior entre cuatro y cinco veces al de SVM radial, esto se invirtió en el test y los tiempos para SVM polinomial fueron la mitad de los de SVM radial. Los tiempos de entrenamiento de SVM radial y ANN se podían comparar, sin embargo, para el test los tiempos de ANN fueron un orden de magnitud menores que los de SVM radial.

Los tiempos para las bases recargas telefónicas y trx. financieras tuvieron un comportamiento similar. Los tiempos de test se mantuvieron prácticamente constantes para CART, C4.5 y ANN entrenados en las diferentes muestras, todos menores a 5 segundos. Aunque los mayores tiempos de entrenamiento fueron para ANN, en el test los tiempos fueron mayores para SVM radial.

En todas las bases los tiempos de test para los modelos SVM aumentaron cuando fueron entrenados en muestras más grandes.

Entren.	E50	E25	E10
SVM_rad	234,0	837,7	3.906,0
SVM_pol	822,5	4.165,5	14.370,4
CART	4,9	12,6	68,1
C4.5	1,0	1,3	3,6
ANN	309,3	599,4	1.009,1

(i)

Test	E50	E25	E10
SVM_rad	1.058	1.555	2.287
SVM_pol	471	838	1.060
CART	39	39	38
C4.5	47	42	43
ANN	74	65	70

(ii)

Tabla 5.11: Tiempos en segundos de: (i) promedio para el entrenamiento y (ii) Testα, base total.

Entren.	E50	E25	E10
SVM_rad	9,8	25,1	84.8
SVM_pol	6,4	27,2	124.7
CART	1,1	2,9	7,6
C4.5	0,7	0,5	1,1
ANN	51,8	110,0	278,7

(i)

Test	E50	E25	E10
SVM_rad	15	15	24
SVM_pol	6	7	10
CART	3	2	3
C4.5	2	3	2
ANN	5	5	4

(ii)

Tabla 5.12: Tiempos en segundos de: (i) promedio para el entrenamiento y (ii) Testβ, base recargas telefónicas.

Entren.	E50	E25	E10
SVM_rad	7,4	19,6	61,5
SVM_pol	6,4	20,3	78,3
CART	1,1	2,1	4,4
C4.5	0,7	0,4	0,8
ANN	41,8	86,4	218,2

(i)

Test	E50	E25	E10
SVM_rad	16	25	33
SVM_pol	8	11	15
CART	3	3	3
C4.5	2	3	3
ANN	4	4	4

(ii)

Tabla 5.13: Tiempos en segundos de: (i) promedio para el entrenamiento y (ii) Testγ, base trx. financieras.

Capítulo 6

Resultados y análisis

Como se mencionó en la sección 5.1 se usaron dos tipos de bases test. En el primer tipo se usó la tasa real de fraude y el segundo una tasa artificial del 0,5 %. En ambas bases test se usaron las mismas transacciones fraudulentas y se varío la cantidad de transacciones normales para alcanzar la tasa de fraude deseada. El resultado que se obtuvo en todos los tipos de modelos, independiente en que base de entrenamiento fue construido, fue que los porcentajes de predicciones correctas e incorrectas eran el mismo para ambas bases test. Esto quiere decir que los modelos son estables, pues al aumentar el n° de transacciones normales los porcentajes de transacciones clasificadas correcta e incorrectamente se mantuvieron. Por esta razón, y para disminuir la gran cantidad de modelos a analizar, sólo se utilizarán las bases test con la tasa real de fraude.

Por confidencialidad no se puede informar la tasa de fraude real, luego las matrices de confusión que se muestran son porcentuales, con respecto a la clasificación verdadera de transacciones. Para que quede claro cómo interpretar las matrices de confusión, en la tabla 6.1 se muestra un ejemplo de cómo se pasa de la matriz de confusión a la matriz de confusión porcentual. En las columnas se tiene la clase verdadera y en las filas la clase predicha, al igual que en la tabla 2.1 (ver sección 2.3). Cabe mencionar que los elementos en la diagonal de la matriz de confusión porcentual son medidas muy conocidas en estadística, el primer elemento de la diagonal se conoce como "Sensibilidad" y el segundo como "Especificidad". La sensibilidad da la precisión en los casos fraudulentos y la especificidad la precisión en los casos normales. Típicamente se observa un *trade-off* (cuando se pierde en una medida se gana en la otra) entre estas medidas.

	P	N
P'	a	b
N'	c	d
	$a+c$	$b+d$

(i)

	P	N
P'	$\frac{a}{a+c}$	$\frac{b}{b+d}$
N'	$\frac{c}{a+c}$	$\frac{d}{b+d}$

(ii)

Tabla 6.1: (i) Matriz de confusión, (ii) Matriz de confusión porcentual.

Los indicadores que se usarán para medir el rendimiento son:

- Precisión global: precisión del modelo, es decir, el porcentaje de casos correctamente clasificados. Toma valores entre 0 y 1, mientras más alto mejor precisión.
- Sensibilidad: precisión en los casos fraudulentos. Toma valores entre 0 y 1, mientras más alto mejor precisión.
- Especificidad: precisión en los casos normales. Toma valores entre 0 y 1, mientras más alto mejor precisión.
- nFP: número de casos normales mal clasificados como fraudulentos (falsos positivos) por cada caso fraudulento correctamente clasificado (verdadero positivo). Toma valores mayores o iguales a 0, mientras más bajo mejor resultado.
- AUC: área bajo la curva ROC (*Receiver Operating Characteristics*). Toma valores entre 0 y 1, mientras más alto mejor clasificación.
- MCC: *Matthews correlation coefficient* es usado para medir la calidad de clasificaciones binarias. Toma valores entre -1 y 1, mientras más alto mejor predicción.
- KS: distancia de Kolmogorov-Smirnov. Toma valores entre 0 y 1, mientras más alto mejor discriminación.

Mayor detalle y las fórmulas de estas medidas se encuentran en el apéndice A. Todas estas medidas, excepto AUC, son generadas a partir de la matriz de confusión. La matriz de confusión se construye bajo la elección de un punto de corte en las probabilidades de predicción, para decidir en base a la probabilidad cuando una transacción se clasifica como fraudulenta. Este corte es típicamente 0,5, es decir, si la probabilidad de una transacción de ser fraudulenta es superior o igual a 0,5, ésta se predice como fraudulenta. El AUC es una medida que se calcula a partir de la curva ROC, la cual es independiente de este punto de corte (para mayor información ver [16]).

Otra forma de medir el desempeño de los modelos, independiente del punto de corte asociado a la matriz de confusión, es el porcentaje de casos fraudulentos capturado para altas probabilidades de predicción. Este se calcula ordenando la base test de manera descendente según la probabilidad de predicción y luego se toma un porcentaje de la parte superior (casos más probable de ser fraudulentos) y se calcula el porcentaje que realmente es fraudulento. El problema con esta medida es que los modelos fueron construidos sólo con variables discretas, así existen varias transacciones con la misma probabilidad de ser fraudulentas (bloques de transacciones), luego los porcentajes con más altas probabilidades son aproximaciones, por ejemplo si se está tomando el 1 % superior, en algunos modelos se puede estar tomando el 0,9 % o el 1,1 % (se encuentra un bloque). En casos más extremos puede que algún porcentaje no aplique para algún modelo, pues existen demasiadas transacciones con la misma probabilidad en ese rango.

En las siguientes secciones se muestran los resultados para las bases total, recargas telefónicas y trx. financieras.

SVM_rad		SVM_pol		CART		C4.5		ANN		Log	
84,5	9,3	84,0	9,6	84,5	9,4	84,6	10,8	85,6	11,3	81,6	12,1
15,5	90,7	16,0	90,4	15,5	90,6	15,4	89,2	14,4	88,7	18,4	87,9

Tabla 6.2: Matrices de confusión porcentual para los modelos entrenados en la base E50, base total.

Modelo	Prec. global	Sensib.	Especif.	nFP	AUC	MCC	KS
SVM_rad	0,907	0,845	0,907	457,2	0,935	0,040	0,752
SVM_pol	0,904	0,840	0,904	478,5	0,930	0,039	0,744
CART	0,906	0,845	0,906	464,4	0,929	0,040	0,751
C4.5	0,892	0,846	0,892	531,3	0,929	0,037	0,738
ANN	0,887	0,856	0,887	549,3	0,939	0,036	0,738
Log	0,879	0,816	0,879	616,1	0,920	0,033	0,696

Tabla 6.3: Medidas para los modelos entrenados en la base E50, base total.

6.1. Base total

6.1.1. Entrenamiento E50

En la tabla 6.2 se tiene la matriz de confusión porcentual para los modelos entrenados en la muestra E50. Los dos modelos con mayor especificidad son SVM radial y CART, pero tienen igual sensibilidad, luego tiene mejor rendimiento el primero. La más alta sensibilidad la posee ANN y luego C4.5. Para ambas medidas el peor resultado lo tiene la regresión logística. Si se mira la tabla 6.3 con las medidas de rendimiento, el peor desempeño en todas las medidas lo tiene la regresión logística. El modelo SVM radial tiene la más alta precisión global, especificidad, MCC y KS y el más bajo nFP. El modelo ANN tiene la más alta sensibilidad y AUC.

En la tabla 6.4 se muestran los porcentajes de casos fraudulentos para las más altas probabilidades de predicción. Al tomar el 0,5 % y 1 % de las probabilidades más altas, el modelo con mejor rendimiento es ANN y al tomar el 5 % y el 10 % el mejor es SVM radial. Pero en estos últimos casos la diferencia con ANN es pequeña, luego en general el modelo con mejor desempeño es ANN. El 0,5 % no aplica para los modelos CART y C4.5.

Modelo	0,5 %	1 %	5 %	10 %
SVM_rad	34,6	51,0	78,4	84,9
SVM_pol	38,5	49,6	77,5	84,3
CART	-	39,8	74,6	84,6
C4.5	-	28,6	75,6	83,3
ANN	44,1	54,8	77,4	84,8
Log	33,4	43,0	68,4	79,0

Tabla 6.4: Porcentajes de casos fraudulentos para las más altas probabilidades, E50 base total.

6.1.2. Entrenamiento E25

En la tabla 6.5 se tiene la matriz de confusión porcentual para los modelos entrenados en la muestra E25. El modelo con mayor especificidad es SVM radial. La más alta sensibilidad la posee C4.5 y luego SVM polinomial, las dos muy cercanas, pero en cuanto a especificidad es mejor el segundo. Luego los modelos SVM tienen el mejor rendimiento. Para ambas medidas nuevamente el peor resultado lo tiene la regresión logística. Si se mira la tabla 6.6 con las medidas de rendimiento, el peor desempeño en todas las medidas lo tiene la regresión logística. El modelo SVM radial tiene la más alta precisión global, especificidad, MCC y el más bajo nFP. El modelo ANN tiene la mayor AUC y el modelo SVM polinomial tiene el mayor KS.

SVM_rad	
77,4	4,1
22,6	95,9

SVM_pol	
78,8	4,4
21,3	95,6

CART	
76,9	4,6
23,1	95,4

C4.5	
78,9	5,4
21,1	94,6

ANN	
77,5	4,5
22,5	95,5

Log	
70,3	5,5
29,8	94,5

Tabla 6.5: Matrices de confusión porcentual para los modelos entrenados en la base E25, base total.

En la tabla 6.7 se muestran los porcentajes de casos fraudulentos para las más altas probabilidades de predicción. Al tomar el 0,5 %, 1 % y 10 % de las probabilidades más altas, el modelo con mejor rendimiento es ANN y al tomar el 5 % existe un empate entre los modelos SVM. En general el modelo con mejor desempeño es ANN. El 0,5 % no aplica para los modelos CART y C4.5.

Modelo	Prec. global	Sensib.	Especif.	nFP	AUC	MCC	KS
SVM_rad	0,959	0,774	0,959	218,7	0,919	0,057	0,733
SVM_pol	0,956	0,788	0,956	233,3	0,929	0,056	0,743
CART	0,954	0,769	0,954	248,4	0,924	0,053	0,723
C4.5	0,945	0,789	0,946	287,6	0,930	0,050	0,734
ANN	0,955	0,775	0,955	240,1	0,940	0,055	0,730
Log	0,945	0,703	0,945	327,5	0,919	0,044	0,647

Tabla 6.6: Medidas para los modelos entrenados en la base E25, base total.

Modelo	0,5 %	1 %	5 %	10 %
SVM_rad	39,5	51,9	79,4	83,6
SVM_pol	34,8	49,4	79,4	84,0
CART	-	52,9	77,9	82,5
C4.5	-	48,6	78,1	81,1
ANN	45,3	57,6	78,6	84,3
Log	37,3	45,0	69,5	78,5

Tabla 6.7: Porcentajes de casos fraudulentos para las más altas probabilidades, E25 base total.

6.1.3. Entrenamiento E10

En la tabla 6.8 se tiene la matriz de confusión porcentual para los modelos entrenados en la muestra E10. Los modelos con mayor especificidad son los SVM, pero el radial tiene mayor sensibilidad. La más alta sensibilidad la posee ANN. Nuevamente para ambas medidas el peor resultado lo tiene la regresión logística. Si se mira la tabla 6.9 con las medidas de rendimiento, el peor desempeño en todas las medidas lo tiene la regresión logística. El modelo SVM radial tiene la más alta precisión global, especificidad, MCC y el más bajo nFP. El modelo ANN tiene la mayor sensibilidad, AUC y KS.

En la tabla 6.10 se muestran los porcentajes de casos fraudulentos para las más altas probabilidades de predicción. Al tomar el 0,5 % y 1 % de las probabilidades más altas, el modelo con mejor rendimiento es SVM radial y al tomar el 5 % y 10 % el modelo con mejor resultado es ANN, pero en los primeros casos las diferencias con ANN son pequeñas, luego en general el modelo con mejor desempeño es ANN.

SVM_rad		SVM_pol		CART		C4.5		ANN		Log	
60,3	1,1	59,6	1,1	63,6	1,5	63,5	1,6	65,4	1,7	54,8	1,8
39,8	98,9	40,4	98,9	36,4	98,5	36,5	98,4	34,6	98,3	45,3	98,2

Tabla 6.8: Matrices de confusión porcentual para los modelos entrenados en la base E10, base total.

Modelo	Prec. global	Sensib.	Especif.	nFP	AUC	MCC	KS
SVM_rad	0,989	0,603	0,989	77,3	0,878	0,087	0,591
SVM_pol	0,988	0,596	0,989	80,0	0,890	0,085	0,585
CART	0,984	0,636	0,985	101,3	0,920	0,078	0,621
C4.5	0,984	0,635	0,984	106,5	0,930	0,076	0,619
ANN	0,983	0,654	0,983	106,9	0,942	0,076	0,637
Log	0,982	0,548	0,982	134,3	0,918	0,062	0,530

Tabla 6.9: Medidas para los modelos entrenados en la base E10, base total.

Modelo	0,5 %	1 %	5 %	10 %
SVM_rad	48,9	58,9	74,8	75,1
SVM_pol	48,5	58,1	74,8	78,0
CART	41,5	56,3	75,8	79,6
C4.5	40,6	54,4	77,6	83,0
ANN	48,1	58,1	78,4	85,1
Log	37,5	44,1	68,3	78,4

Tabla 6.10: Porcentajes de casos fraudulentos para las más altas probabilidades, E10 base total.

6.2. Base recargas telefónicas

6.2.1. Entrenamiento E50

En la tabla 6.11 se tiene la matriz de confusión porcentual para los modelos entrenados en la muestra E50. El modelo con mayor sensibilidad es CART y le siguen igualados los modelos SVM y ANN, pero de éstos el con mayor especificidad es SVM polinomial y posee la especificidad más alta de todos. Si se mira la tabla 6.12 con las medidas de rendimiento, el modelo SVM polinomial tiene la más alta precisión global, especificidad, MCC, KS y el más bajo nFP. El modelo ANN tiene la mayor AUC.

En la tabla 6.13 se muestran los porcentajes de casos fraudulentos para las más altas probabilidades de predicción. En general el modelo con mejor desempeño es ANN, sólo para el 0,5 % no es el mejor, pero está a 0,4 % de los mayores. El 0,5 % no aplica para los modelos CART y C4.5 y el 10 % no aplica para CART.

SVM_rad

88,6	6,3
11,4	93,7

SVM_pol

88,6	3,2
11,4	96,8

CART

89,0	4,7
11,0	95,3

C4.5

87,6	3,7
12,4	96,3

ANN

88,6	4,7
11,4	95,3

Log

88,6	6,7
11,4	93,3

Tabla 6.11: Matrices de confusión porcentual para los modelos entrenados en la base E50, base recargas.

Modelo	Prec. global	Sensib.	Especif.	nFP	AUC	MCC	KS
SVM_rad	0,937	0,886	0,937	71,5	0,946	0,106	0,822
SVM_pol	0,968	0,886	0,968	35,6	0,952	0,152	0,854
CART	0,953	0,890	0,953	53,1	0,931	0,124	0,843
C4.5	0,963	0,876	0,963	42,5	0,938	0,138	0,839
ANN	0,953	0,886	0,953	52,7	0,961	0,125	0,839
Log	0,933	0,886	0,933	75,9	0,955	0,103	0,818

Tabla 6.12: Medidas para los modelos entrenados en la base E50, base recargas.

Modelo	0,5 %	1 %	5 %	10 %
SVM_rad	40,5	61,4	87,1	90,5
SVM_pol	67,1	78,1	89,0	90,0
CART	-	3,3	89,0	-
C4.5	-	73,3	89,0	90,5
ANN	71,4	81,4	88,6	90,5
Log	63,8	80,5	87,1	88,6

Tabla 6.13: Porcentajes de casos fraudulentos para las más altas probabilidades, E50 base recargas.

6.2.2. Entrenamiento E25

En la tabla 6.14 se tiene la matriz de confusión porcentual para los modelos entrenados en la muestra E25. Cuatro modelos comparten la más alta sensibilidad, dentro de los cuales se encuentra el modelo con mayor especificidad: SVM radial, luego éste presenta el mejor rendimiento para ambas medidas. Si se mira la tabla 6.15 con las medidas de rendimiento, el modelo SVM radial tiene la más alta precisión global, especificidad, MCC, KS y el más bajo nFP. El modelo ANN tiene la mayor AUC.

En la tabla 6.16 se muestran los porcentajes de casos fraudulentos para las más altas probabilidades de predicción. Para el 0,5 % tiene mejor rendimiento la regresión logística y ANN, para el 1 % empatan SVM polinomial y ANN, para el 5 % CART y para el 10 % empatan CART y C4.5. En términos generales el modelo con mejor desempeño es ANN. El 0,5 % no aplica para los modelos CART y C4.5, el 5 % no aplica para C4.5 y el 10 % no aplica para los modelos SVM .

SVM_rad

85,2	1,6
14,8	98,4

SVM_pol

84,8	1,7
15,2	98,3

CART

84,8	2,1
15,2	97,9

C4.5

85,2	2,4
14,8	97,6

ANN

85,2	2,1
14,8	97,9

Log

85,2	2,3
14,8	97,7

Tabla 6.14: Matrices de confusión porcentual para los modelos entrenados en la base E25, base recargas.

Modelo	Prec. global	Sensib.	Especif.	nFP	AUC	MCC	KS
SVM_rad	0,983	0,852	0,984	19,3	0,939	0,203	0,836
SVM_pol	0,983	0,848	0,983	20,2	0,933	0,198	0,830
CART	0,979	0,848	0,979	25,0	0,943	0,178	0,826
C4.5	0,976	0,852	0,976	27,7	0,943	0,170	0,829
ANN	0,979	0,852	0,979	24,2	0,956	0,181	0,832
Log	0,976	0,852	0,977	27,5	0,954	0,170	0,829

Tabla 6.15: Medidas para los modelos entrenados en la base E25, base recargas.

Modelo	0,5 %	1 %	5 %	10 %
SVM_rad	71,4	82,9	87,6	-
SVM_pol	69,5	81,9	88,6	-
CART	-	72,4	89,5	91,0
C4.5	-	78,6	-	91,0
ANN	73,3	82,9	87,1	88,6
Log	73,3	81,4	87,1	88,6

Tabla 6.16: Porcentajes de casos fraudulentos para las más altas probabilidades, E25 base recargas.

6.2.3. Entrenamiento E10

En la tabla 6.17 se tiene la matriz de confusión porcentual para los modelos entrenados en la muestra E10. Todos los modelos tienen una alta y parecida especificidad, tomando valores entre 0,989 y 0,992, es decir, existe una diferencia de sólo 0,003 entre el menor y mayor valor. El modelo con mejor sensibilidad es CART, seguido de la regresión logística. Si se mira la tabla 6.18 con las medidas de rendimiento, el modelo SVM radial tiene el más bajo nFP y el más alto MCC. El modelo ANN tiene la mayor AUC y CART el más alto KS.

En la tabla 6.19 se muestran los porcentajes de casos fraudulentos para las más altas probabilidades de predicción. Para el 0,5 % tiene mejor rendimiento la regresión logística, para el 1 % ANN y para el 5 % y 10 % CART. En general el modelo con mejor desempeño es CART. El 0,5 % no aplica para C4.5, el 5 % no aplica para C4.5 y SVM polinomial y el 10 % no aplica para SVM radial y C4.5.

SVM_rad	
80,0	0,8
20,0	99,2

SVM_pol	
80,5	1,0
19,5	99,0

CART	
82,4	1,1
17,6	98,9

C4.5	
80,0	1,0
20,0	99,0

ANN	
81,4	1,0
18,6	99,0

Log	
81,9	0,9
18,1	99,1

Tabla 6.17: Matrices de confusión porcentual para los modelos entrenados en la base E10, base recargas.

Modelo	Prec. global	Sensib.	Especif.	nFP	AUC	MCC	KS
SVM_rad	0,991	0,800	0,992	10,5	0,921	0,263	0,792
SVM_pol	0,990	0,805	0,990	12,1	0,903	0,246	0,795
CART	0,989	0,824	0,989	13,7	0,947	0,235	0,813
C4.5	0,990	0,800	0,990	12,2	0,906	0,244	0,790
ANN	0,990	0,814	0,990	12,2	0,956	0,247	0,804
Log	0,991	0,819	0,991	11,0	0,954	0,259	0,810

Tabla 6.18: Medidas para los modelos entrenados en la base E10, base recargas.

Modelo	0,5%	1%	5%	10%
SVM_rad	71,9	81,0	87,1	-
SVM_pol	62,9	80,5	-	83,3
CART	74,8	82,4	89,5	91,0
C4.5	-	80,0	-	-
ANN	74,8	82,9	88,1	90,0
Log	75,7	81,9	87,6	89,0

Tabla 6.19: Porcentajes de casos fraudulentos para las más altas probabilidades, E10 base recargas.

6.3. Base trx. financieras

6.3.1. Entrenamiento E50

En la tabla 6.20 se tiene la matriz de confusión porcentual para los modelos entrenados en la muestra E50. Los modelos con la mayor sensibilidad son los árboles de decisión, pero tienen las más bajas especificidades. El modelo con mayor especificidad es SVM radial. Si se mira la tabla 6.21 con las medidas de rendimiento, el modelo SVM radial tiene la más alta precisión global, especificidad, MCC, KS y el más bajo nFP. El modelo ANN tiene la mayor AUC, pero es sólo 0,001 mayor que la de SVM radial.

En la tabla 6.22 se muestran los porcentajes de casos fraudulentos para las más altas probabilidades de predicción. Para el 0,5% y 1% tiene mejor rendimiento la regresión logística y para el 5% y 10% ANN. El 0,5% y 1% no aplican para CART y C4.5, el 5% no aplica para CART.

SVM_rad

79,8	19,2
20,2	80,8

SVM_pol

79,1	19,9
20,9	80,1

CART

81,4	22,2
18,6	77,8

C4.5

81,4	23,7
18,6	76,3

ANN

78,3	21,4
21,7	78,6

Log

78,3	20,8
21,7	79,2

Tabla 6.20: Matrices de confusión porcentual para los modelos entrenados en la base E50, base trx. financieras.

Modelo	Prec. global	Sensib.	Especif.	nFP	AUC	MCC	KS
SVM_rad	0,808	0,798	0,808	401,1	0,867	0,038	0,606
SVM_pol	0,801	0,791	0,801	419,2	0,854	0,036	0,592
CART	0,778	0,814	0,778	455,0	0,854	0,035	0,592
C4.5	0,763	0,814	0,763	485,4	0,843	0,033	0,577
ANN	0,786	0,783	0,786	455,3	0,868	0,034	0,569
Log	0,792	0,783	0,792	441,8	0,864	0,035	0,575

Tabla 6.21: Medidas para los modelos entrenados en la base E50, base trx. financieras.

Modelo	0,5 %	1 %	5 %	10 %
SVM_rad	10,9	21,7	51,9	58,1
SVM_pol	9,3	18,6	44,2	60,5
CART	-	-	-	61,2
C4.5	-	-	36,4	50,4
ANN	12,4	24,0	49,6	65,1
Log	15,5	26,4	48,1	63,6

Tabla 6.22: Porcentajes de casos fraudulentos para las más altas probabilidades, E50 base trx. financieras.

6.3.2. Entrenamiento E25

En la tabla 6.23 se tiene la matriz de confusión porcentual para los modelos entrenados en la muestra E25. El modelo con la mayor sensibilidad es CART, pero tiene la más baja especificidad. El modelo con mayor especificidad es SVM radial. Los modelos C4.5 y ANN tienen una sensibilidad menor a 0,5, es decir, menos de la mitad de los casos fraudulentos se clasifican correctamente, hasta ahora todos los modelos han tenido una sensibilidad superior a 0,5. Si se mira la tabla 6.24 con las medidas de rendimiento, el modelo SVM radial tiene la más alta precisión global, especificidad y el más bajo nFP. El modelo ANN tiene la mayor AUC y CART tiene la más alta sensibilidad y KS. Estos tres modelos tienen el más alto MCC.

En la tabla 6.25 se muestran los porcentajes de casos fraudulentos para las más altas probabilidades de predicción. Para el 0,5 % y 10 % tiene mejor rendimiento SVM radial, para el 1 % y 5 % ANN. El 0,5 % no aplica para CART y C4.5.

SVM_rad	
54,3	5,9
45,7	94,1

SVM_pol	
52,7	6,3
47,3	93,7

CART	
59,7	7,1
40,3	92,9

C4.5	
47,3	6,1
52,7	93,9

ANN	
58,9	7,0
41,1	93,0

Log	
52,7	7,0
47,3	93,0

Tabla 6.23: Matrices de confusión porcentual para los modelos entrenados en la base E25, base trx. financieras.

Modelo	Prec. global	Sensib.	Especif.	nFP	AUC	MCC	KS
SVM_rad	0,941	0,543	0,941	181,1	0,863	0,050	0,484
SVM_pol	0,937	0,527	0,937	197,7	0,868	0,047	0,465
CART	0,929	0,597	0,929	198,3	0,832	0,050	0,526
C4.5	0,939	0,473	0,939	214,9	0,827	0,042	0,412
ANN	0,929	0,589	0,930	199,2	0,870	0,050	0,519
Log	0,929	0,527	0,930	222,7	0,866	0,044	0,457

Tabla 6.24: Medidas para los modelos entrenados en la base E25, base trx. financieras.

Modelo	0,5 %	1 %	5 %	10 %
SVM_rad	20,9	25,6	51,2	69,0
SVM_pol	19,4	24,8	49,6	66,7
CART	-	23,3	46,5	63,6
C4.5	-	17,1	45,7	52,7
ANN	14,0	26,4	51,9	65,9
Log	19,4	24,8	48,8	61,2

Tabla 6.25: Porcentajes de casos fraudulentos para las más altas probabilidades, E25 base trx. financieras.

6.3.3. Entrenamiento E10

En la tabla 6.26 se tiene la matriz de confusión porcentual para los modelos entrenados en la muestra E10. El modelo con mayor especificidad es C4.5, pero tiene la más baja sensibilidad. El modelo con mayor sensibilidad es ANN. Todos los modelos tienen una sensibilidad menor a 0,33, es decir, menos del tercio de las transacciones fraudulentas son predichas correctamente. Si se mira la tabla 6.27 con las medidas de rendimiento, el modelo SVM polinomial tiene la más alta precisión global, especificidad y el más bajo nFP. El modelo ANN tiene la mayor sensibilidad, AUC y KS. El modelo SVM radial tiene el mayor MCC.

En la tabla 6.28 se muestran los porcentajes de casos fraudulentos para las más altas probabilidades de predicción. Para el 0,5 % y 1 % tiene mejor rendimiento SVM polinomial y para el 5 % y 10 % ANN. El 5 % no aplica para C4.5 y el 10 % no aplica para CART y C4.5.

SVM_rad	
27,1	0,9
72,9	99,1

SVM_pol	
24,0	0,7
76,0	99,3

CART	
26,4	1,1
73,6	98,9

C4.5	
23,3	0,8
76,7	99,2

ANN	
32,6	1,5
67,4	98,5

Log	
31,0	1,3
69,0	98,7

Tabla 6.26: Matrices de confusión porcentual para los modelos entrenados en la base E10, base trx. financieras.

Modelo	Prec. global	Sensib.	Especif.	nFP	AUC	MCC	KS
SVM_rad	0,991	0,271	0,991	53,4	0,794	0,069	0,263
SVM_pol	0,992	0,240	0,993	49,6	0,795	0,067	0,233
CART	0,988	0,264	0,989	72,0	0,775	0,058	0,252
C4.5	0,991	0,233	0,992	58,3	0,656	0,061	0,224
ANN	0,984	0,326	0,985	79,0	0,870	0,061	0,310
Log	0,986	0,310	0,987	72,0	0,868	0,063	0,297

Tabla 6.27: Medidas para los modelos entrenados en la base E10, base trx. financieras.

Modelo	0,5%	1%	5%	10%
SVM_rad	21,7	27,1	41,9	51,9
SVM_pol	20,9	24,8	41,9	46,5
CART	17,8	26,4	46,5	-
C4.5	21,7	23,3	-	-
ANN	20,2	23,3	50,4	66,7
Log	20,9	25,6	50,4	59,7

Tabla 6.28: Porcentajes de casos fraudulentos para las más altas probabilidades, E10 base trx. financieras.

6.4. Comparación modelos

En esta sección se analiza el efecto de construir modelos en diferentes muestras de entrenamiento. Para esto se promediaron las medidas de rendimiento de los modelos para las muestras E50, E25 y E10 y también se calculó la diferencia relativa entre los valores mejores y peores de cada medida. Los resultados para las tres bases se muestran en la tabla 6.29 y las diferencias relativas en las tablas 6.30 - 6.32. A medida que aumenta el tamaño de la muestra y disminuye la proporción de transacciones fraudulentas (desde E50 a E10) aumenta la precisión global y la especificidad, pues se predice mejor las transacciones normales (la clase de mayor tamaño) y disminuye la sensibilidad, pues se predice peor las transacciones fraudulentas (*trade-off*). Como consecuencia disminuye el nFP y aumenta el MCC, lo que indica una mejor clasificación. Además disminuye el AUC, lo que representa una peor clasificación, independiente del punto de corte (visto al principio del capítulo) y disminuye el KS, lo que indica menor poder discriminante.

En la base total, a medida que disminuye la proporción de transacciones fraudulentas en el entrenamiento, la diferencia relativa entre el mejor y peor valor para precisión global y especificidad disminuye, pero para las restantes medidas este valor aumenta. Esta diferencia para la especificidad y precisión global es pequeña, menor al 4 %, para las tres muestras de entrenamiento. Para E50 la diferencia en sensibilidad es 4,9 %, pero aumenta hasta 19,3 % en E10, luego en esta última se tiene una diferencia significativa entre el mejor y peor valor. Para nFP y MCC las diferencias son superiores al 20 %, siendo sobre el 40 % para E10. Las diferencias en AUC son menores al 8 % y para el KS varía entre 8 % y 17,2 %. Exceptuando la precisión global y especificidad, todas las medidas tienen una mayor variación en la muestra con menor proporción de transacciones fraudulentas.

Total	Prec. global	Sensib.	Especif.	nFP	AUC	MCC	KS
Prom. E50	0,896	0,841	0,896	516,1	0,930	0,038	0,737
Prom. E25	0,952	0,766	0,953	259,3	0,927	0,053	0,718
Prom. E10	0,985	0,611	0,985	101,1	0,913	0,077	0,597

Recargas	Prec. global	Sensib.	Especif.	nFP	AUC	MCC	KS
Prom. E50	0,951	0,885	0,951	55,2	0,947	0,125	0,836
Prom. E25	0,979	0,851	0,980	24,0	0,945	0,183	0,830
Prom. E10	0,990	0,810	0,990	12,0	0,931	0,249	0,801

Trx. finan.	Prec. global	Sensib.	Especif.	nFP	AUC	MCC	KS
Prom. E50	0,788	0,797	0,788	443,0	0,858	0,035	0,585
Prom. E25	0,934	0,543	0,934	202,3	0,854	0,047	0,477
Prom. E10	0,989	0,274	0,990	64,1	0,793	0,063	0,263

Tabla 6.29: Promedio de las medidas de los modelos, en las bases total, recargas y trx. financieras.

En la base recargas telefónicas, la diferencia relativa entre el mejor y peor valor para precisión global, especificidad, sensibilidad, AUC y KS es pequeña, todas menores al 6 %. En este caso no se encuentran grandes diferencias en los modelos en cuanto especificidad y sensibilidad, pero para otras medidas como nFP y MCC las diferencias son superiores. El nFP varía entre el 23,4 % y 53,1 %, mientras que el MCC entre 11,9 % y el 47,6 %.

En la base trx. financieras, se tiene la misma tendencia que en la base total. La diferencia relativa para la especificidad y precisión global es pequeña, menor al 6 %, para las tres muestras de entrenamiento. Para E50 la diferencia en sensibilidad es 4 %, pero aumenta hasta 39,9 % en E10, luego en esta última se tiene una diferencia significativa entre el mejor y peor valor y el valor es mayor que en la base total. Para nFP y MCC las diferencias son superiores al 15 %, siendo sobre el 35 % para el nFP en E10. Las diferencias en AUC son pequeñas para E50 y E25, pero en E10 es 32,6 %. El KS varía entre 6,5 % y 38,4 %. En esta a base a medida que se disminuye la proporción de transacciones fraudulentas las medidas tienen grandes variaciones, mayores en general que la base total.

Con la tabla 6.29 se pueden comparan los modelos entre las bases. Claramente se obtienen mejores resultados en la base recargas telefónicas y entre la base total y trx. financieras se tienen mejores resultados en la primera, excepto para la medida nFP. Este es un análisis macro, pues son los promedios de los indicadores, en la sección 6.6 se hará una comparación con más detalle.

E50	Prec. global	Sensib.	Especif.	nFP	AUC	MCC	KS
Peor	0,879	0,816	0,879	616,1	0,920	0,033	0,696
Mejor	0,907	0,856	0,907	457,2	0,939	0,040	0,752
Dif. rel.	3,2%	4,9%	3,2%	-25,8%	2,1%	21,2%	8,0%

E25	Prec. global	Sensib.	Especif.	nFP	AUC	MCC	KS
Peor	0,945	0,703	0,945	327,5	0,919	0,044	0,647
Mejor	0,959	0,789	0,959	218,7	0,940	0,057	0,743
Dif. rel.	1,5%	12,2%	1,5%	-33,2%	2,3%	29,5%	14,8%

E10	Prec. global	Sensib.	Especif.	nFP	AUC	MCC	KS
Peor	0,982	0,548	0,982	134,3	0,878	0,062	0,530
Mejor	0,989	0,654	0,989	77,3	0,942	0,087	0,621
Dif. rel.	0,7%	19,3%	0,7%	-42,4%	7,3%	40,3%	17,2%

Tabla 6.30: Diferencia relativa entre el mejor y peor modelo según cada indicador, base total.

E50	Prec. global	Sensib.	Especif.	nFP	AUC	MCC	KS
Peor	0,933	0,876	0,933	75,9	0,931	0,103	0,818
Mejor	0,968	0,890	0,968	35,6	0,961	0,152	0,854
Dif. rel.	3,8%	1,6%	3,8%	-53,1%	3,2%	47,6%	4,4%

E25	Prec. global	Sensib.	Especif.	nFP	AUC	MCC	KS
Peor	0,976	0,848	0,976	27,7	0,933	0,170	0,826
Mejor	0,983	0,852	0,984	19,3	0,956	0,203	0,836
Dif. rel.	0,7%	0,5%	0,8%	-30,3%	2,5%	19,4%	1,2%

E10	Prec. global	Sensib.	Especif.	nFP	AUC	MCC	KS
Peor	0,989	0,800	0,989	13,7	0,903	0,235	0,790
Mejor	0,991	0,824	0,992	10,5	0,956	0,263	0,813
Dif. rel.	0,2%	3,0%	0,3%	-23,4%	5,9%	11,9%	2,9%

Tabla 6.31: Diferencia relativa entre el mejor y peor modelo según cada indicador, base recargas.

E50	Prec. global	Sensib.	Especif.	nFP	AUC	MCC	KS
Peor	0,763	0,783	0,763	485,4	0,843	0,033	0,569
Mejor	0,808	0,814	0,808	401,1	0,868	0,038	0,606
Dif. rel.	5,9%	4,0%	5,9%	-17,2%	3,0%	15,2%	6,5%

E25	Prec. global	Sensib.	Especif.	nFP	AUC	MCC	KS
Peor	0,929	0,473	0,929	222,7	0,827	0,042	0,412
Mejor	0,941	0,597	0,941	181,1	0,870	0,050	0,526
Dif. rel.	1,3%	26,2%	1,3%	-18,3%	5,2%	19,0%	27,7%

E10	Prec. global	Sensib.	Especif.	nFP	AUC	MCC	KS
Peor	0,984	0,233	0,985	79,0	0,656	0,058	0,224
Mejor	0,992	0,326	0,993	49,6	0,87	0,069	0,310
Dif. rel.	0,8%	39,9%	0,8%	-37,2%	32,6%	19,0%	38,4%

Tabla 6.32: Diferencia relativa entre el mejor y peor modelo según cada indicador, base trx. financieras.

6.5. Evaluación económica

Para evaluar el impacto económico de los modelos se procederá de la siguiente manera. Se ordena la base test de forma descendente según la probabilidad de predicción y luego se toma un porcentaje de la parte superior (casos más probable de ser fraudulentos) y se calcula el monto total de transacciones fraudulentas y normales. Expertos del negocio definieron un 10 % de ganancia con respecto a la venta, luego un 10 % del monto de transacciones normales es ingreso neto, en cambio el monto fraudulento se considera pérdida. Para medir el impacto económico se supondrá que el porcentaje escogido de transacciones más probables de ser fraudulentas se detendrán, es decir, se dejará de percibir el ingreso neto y no se perderá el monto fraudulento, luego el indicador económico (IE) será:

$$IE = (\text{monto fraudulento}) - 10\,\%(\text{monto normal}) = (\text{monto fraudulento}) - (\text{ingreso neto}) \tag{6.1}$$

Si IE es positivo convendría detener esas transacciones y el modelo entrega buenos resultados económicos, si IE es negativo sería mejor dejar pasar esas transacciones y el modelo no tiene un buen impacto económico. Análogamente a los porcentajes de casos fraudulentos para las más altas probabilidades de predicción, explicado al comienzo de este capítulo, existen modelos para los cuales algún porcentaje no aplica.

Los porcentajes escogidos para calcular el IE son: 0,1 %, 0,2 %, 0,5 % y 1 %. Estos son pequeños, pues son transacciones que en la práctica se detendrían y no pueden ser muchas (se afecta a los clientes del negocio). Expertos del negocio recomendaron no parar más del 1 % de las transacciones.

6.5.1. Impacto económico en la base total

En la tabla 6.33 se muestran los valores del indicador económico para la base total. Para todos los modelos, independiente del porcentaje usado, se tiene un impacto económico negativo, por lo que estos modelos generarían pérdidas en la práctica. Esto muestra que un modelo que toma todos los tipos de transacciones no es efectivo, por la variabilidad de fraude. Sólo se puede analizar en que modelos se pierde menos dinero. Todos los modelos tienen menor IE para el 0,1 %. Para los modelos construidos en E50, SVM radial tiene los mayores indicadores económicos (IE's), excepto para el 1 %. Para E25, ANN tiene los mayores IE's en 0,1 % y 0,2 % y los segundos en 0,5 % y 1 %. Finalmente, para E10, C4.5 tiene los mayores IE's, excepto para 0,5 %. Sólo en los árboles de decisión hay porcentajes que no aplican.

E50	0,1 %	0,2 %	0,5 %	1 %
SVM_rad	-23	-41	-214	-364
SVM_pol	-38	-69	-315	-488
CART	-	-	-	-355
C4.5	-	-	-	-191
ANN	-67	-200	-305	-420
Log	-51	-88	-306	-687

E25	0,1 %	0,2 %	0,5 %	1 %
SVM_rad	-20	-44	-122	-739
SVM_pol	-19	-34	-86	-524
CART	-51	-	-	-373
C4.5	-	-106	-	-171
ANN	-17	-34	-91	-323
Log	-47	-75	-272	-728

E10	0,1 %	0,2 %	0,5 %	1 %
SVM_rad	-16	-39	-196	-324
SVM_pol	-18	-52	-205	-363
CART	-	-50	-91	-290
C4.5	-10	-18	-219	-281
ANN	-20	-51	-117	-393
Log	-43	-73	-271	-443

Tabla 6.33: IE en millones de pesos para los modelos de la base total.

6.5.2. Impacto económico en la base recargas telefónicas

En la tabla 6.34 se muestran los valores del indicador económico para la base recargas telefónicas. La gran mayoría de los modelos tienen un impacto económico positivo, esto quiere decir que generarían ganancias en la práctica. Los mayores IE's se obtienen con los modelos ANN para el 0,2 %. El mayor IE se tiene con el modelo ANN construido en la muestra E10. Nuevamente, en los árboles de decisión hay porcentajes que no aplican.

E50	0,1 %	0,2 %	0,5 %	1 %
SVM_rad	519	234	287	-96
SVM_pol	1.099	1.068	1.141	396
CART	-	-	-	-1.009
C4.5	-	-	-	-31
ANN	915	1.458	1.220	429
Log	1.073	1.104	837	488

E25	0,1 %	0,2 %	0,5 %	1 %
SVM_rad	1.132	1.423	1.236	486
SVM_pol	1.105	1.028	1.175	449
CART	-	-	-	-169
C4.5	-	-381	-	192
ANN	940	1.471	1.328	510
Log	1.158	1.187	1.328	519

E10	0,1 %	0,2 %	0,5 %	1 %
SVM_rad	846	865	1.281	478
SVM_pol	908	897	1.005	523
CART	956	1.098	1.408	365
C4.5	-	899	-	260
ANN	1.158	1.641	1.376	485
Log	1.152	1.238	1.392	567

Tabla 6.34: IE en miles de pesos para los modelos de la base recargas telefónicas.

6.5.3. Impacto económico en la base trx. financieras

En la tabla 6.35 se muestran los valores del indicador económico para la base trx. financieras. Para todos los modelos, independiente del porcentaje usado, se tiene un impacto económico negativo, por lo que estos modelos generarían pérdidas en la práctica, aunque en este caso se considera un sólo tipo de transacción. Esto muestra que aislando un tipo de transacción no siempre se obtienen buenos resultados, como los obtenidos en recargas telefónicas. Se puede analizar en que modelos se pierde menos dinero. Todos los modelos tienen menor IE para el 0,1 %. Para los modelos construidos en E50, ANN tiene los mayores IE's, excepto para el 1 % donde tiene el segundo mayor. Para E25, ANN tiene los mayores IE's en 0,2 % y 0,5 % y los segundos en 0,1 % y 1 %. Finalmente, para E10, C4.5 tiene los mayores IE's, excepto para 0,1 %, donde no aplica. Nuevamente, en los árboles de decisión hay porcentajes que no aplican.

E50	0,1 %	0,2 %	0,5 %	1 %
SVM_rad	-5	-9	-26	-47
SVM_pol	-9	-15	-27	-70
CART	-	-	-	-
C4.5	-	-	-	-
ANN	-5	-7	-20	-51
Log	-5	-8	-24	-69

E25	0,1 %	0,2 %	0,5 %	1 %
SVM_rad	-7	-11	-16	-37
SVM_pol	-3	-11	-28	-46
CART	-	-	-	-64
C4.5	-	-	-	-45
ANN	-4	-5	-12	-39
Log	-3	-7	-23	-78

E10	0,1 %	0,2 %	0,5 %	1 %
SVM_rad	-4	-8	-37	-65
SVM_pol	-4	-8	-32	-61
CART	-	-	-32	-62
C4.5	-	-6	-24	-26
ANN	-6	-15	-36	-68
Log	-4	-9	-28	-75

Tabla 6.35: IE en millones de pesos para los modelos de la base trx. financieras.

6.6. Comparación con la base total

Uno de los objetivos es medir cuanto mejoran los modelos al enfocarse en sólo un rubro, con respecto al modelo en la base total. Para hacer esto se aisló de la base Testα las transacciones de los rubros recargas telefónicas y trx. financieras. Con esto se crearon dos nuevas bases test para recargas y trx. financieras, denotadas Testα_rec y Testα_fin, respectivamente. En estas bases no están exactamente las mismas transacciones que en las bases Testβ y Testγ, pero si tienen la misma tasa de fraude y por la estabilidad de los modelos (ver primer párrafo de este capítulo) si son comparables los resultados de éstos. En un principio se pensó en aplicar el modelo de la base total en las bases Testβ y Testγ, pero esto arrojó errores en R en modelos como los SVMs, ya que no se tiene la misma estructura en las variables de entrada. Por otro lado la forma escogida es más simple de realizar y además no incluye transacciones usadas en el entrenamiento de los modelos.

Para no mostrar nuevamente los resultados obtenidos, se seleccionarán dos modelos para cada muestra de entrenamiento. Para la base recargas telefónicas se usarán SVM radial y ANN de las tres muestras, pues el primero tuvo en general los mejores resultados para la base total y ANN tuvo los mejores IE's para la base recargas telefónicas. Para la base trx. financieras también se usarán SVM radial de las tres muestras, además para E50 y E25 se usarán ANN y para E10 se usará C4.5, pues tienen los mejores IE's para la base trx. financieras.

Las medidas usadas para comparar los modelos serán: sensibilidad, especificidad, nFP, KS y el IE para el 0,2 %.

Testα_rec	Sensib.	Especif.	nFP	KS	IE(M$)
SVM_radial_E50	0,934	0,796	216,7	0,730	112
SVM_radial_E25	0,888	0,943	63,8	0,831	1.062
SVM_radial_E10	0,837	0,986	16,3	0,823	-
ANN_E50	0,929	0,815	198,3	0,743	885
ANN_E25	0,878	0,958	47,4	0,836	1.125
ANN_E10	0,857	0,978	25,7	0,835	1.438

Recargas	Sensib.	Especif.	nFP	KS	IE
SVM_radial_E50	0,886	0,937	71,5	0,822	234
SVM_radial_E25	0,852	0,984	19,3	0,836	1.423
SVM_radial_E10	0,800	0,992	10,5	0,792	865
ANN_E50	0,886	0,953	52,7	0,839	1.458
ANN_E25	0,852	0,979	24,2	0,832	1.471
ANN_E10	0,814	0,990	12,2	0,804	1.641

Tabla 6.36: Comparación modelos de las bases total aplicado a recargas telefónicas y recargas telefónicas.

6.6.1. Recargas telefónicas

En la tabla 6.36 se tienen las medidas para algunos modelos de la base total aplicados a las recargas telefónicas (base Testα_rec) y los modelos en la base recargas telefónicas. Los modelos de la base recargas telefónicas superan en todas las medidas a los modelos en Testα_rec, excepto para la sensibilidad y en algunos casos el KS. Esto quiere decir que el fraude es predicho mejor con modelos de la base total, pero predice peor las transacciones normales, como éstas transacciones son mayoría, hace que el resto de las medidas sean mejores para los modelos de la base recargas y por lo tanto se tenga un mejor desempeño en estos últimos. El número de falsos negativos por cada verdadero positivo (nFP) se reduce entre un 35 % y 75 % para éstos modelos y el indicador económico (IE) aumenta entre un 14 % y 109 %. Luego es mejor aislar este tipo de transacciones y aplicar un modelo sobre éstas.

6.6.2. Transacciones financieras

En la tabla 6.37 se tienen las medidas para algunos modelos de la base total aplicados a las transacciones financieras (base Testα_fin) y los modelos en la base trx. financieras. Los modelos de la base trx. financieras tienen mejor especificidad y nFP y tiene peor sensibilidad. Esto quiere decir que el fraude es predicho mejor con modelos de la base total, pero predice peor las transacciones normales, como éstas transacciones son mayoría, hace que el nFP sea más alto en éstos. El nFP se reduce entre un 49 % y 80 % para los modelos de la base trx. financieras. El KS en algunos casos es mejor en una base y en otros en la otra, luego no da claridad de que base es mejor. El IE es negativo, luego no se analizará. Igualmente es mejor aislar este tipo de transacciones y aplicar un modelo sobre éstas, pero no es tanta la mejora como en recargas telefónicas.

Testα_fin	Sensib.	Especif.	nFP	KS	IE(MM$)
SVM_radial_E50	0,936	0,503	889,0	0,439	-7
SVM_radial_E25	0,816	0,778	455,8	0,594	-3
SVM_radial_E10	0,408	0,944	229,2	0,352	- 5
ANN_E50	0,928	0,503	897,7	0,431	-9
ANN_E25	0,816	0,753	507,4	0,569	-4
C4.5_E10	0,448	0,924	285,2	0,372	-2

Trx. finan.	Sensib.	Especif.	nFP	KS	IE
SVM_radial_E50	0,798	0,808	401,1	0,606	-9
SVM_radial_E25	0,543	0,941	181,1	0,484	-11
SVM_radial_E10	0,271	0,991	53,4	0,263	-8
ANN_E50	0,783	0,786	455,3	0,569	-7
ANN_E25	0,589	0,930	199,2	0,519	-5
C4.5_E10	0,233	0,992	58,3	0,224	-6

Tabla 6.37: Comparación modelos de las bases total aplicado a trx.financieras y trx. financieras.

Capítulo 7

Conclusiones

Uno de los principales problemas en detección de fraude es el gran desbalance del número de transacciones fraudulentas y normales: se tiene una cantidad muy superior de transacciones normales con respecto a la cantidad de transacciones fraudulentas. Para construir los modelos se usaron tres bases con una proporción de transacciones fraudulentas más adecuada, a saber, 50 %, 25 % y 10 %. La primera base es neutra, en el sentido de que hay igual cantidad de transacciones fraudulentas y normales. Las otras en cambio, intentan asemejarse a la realidad, con un menor número de transacciones fraudulentas. En las bases con menor proporción de transacciones fraudulentas los modelos tuvieron una mejor predicción en las transacciones normales (especificidad) y una peor predicción en las fraudulentas (sensibilidad), esto produce una mejor clasificación general, pues la clase mayor tiene mejor predicción. Cuando hay más transacciones normales los modelos se enfocan más en éstas, lo que implica una predicción del fraude algo más baja pero con menos equivocaciones en la predicción global.

Al disminuir la proporción de transacciones fraudulentas en las bases de entrenamiento, se podría encontrar una proporción óptima de transacciones fraudulentas y normales, que se tradujera en una mejor clasificación general de los modelos. En este trabajo se llegó al 10 % de transacciones fraudulentas, al disminuir este valor es probable que mejoren la predicción y los falsos negativos.

Una solución propuesta para detectar fraude fue aislar un tipo de transacciones y aplicar los modelos sólo en éstas. Se eligieron tres agrupaciones de rubros por el número de transacciones y montos de fraudulento: centros de pago, recargas telefónicas y transacciones financieras.

En recargas telefónicas se obtuvieron los mejores resultados y fue el único tipo en el que los modelos tuvieron un impacto económico positivo, esto pues se identificó un claro patrón de fraude: varias transacciones seguidas hasta alcanzar el monto máximo diario permitido. Para identificar esto en los modelos fue importante la variable acumuladora diaria.

En transacciones financieras no se encontró un patrón de fraude, las transacciones normales y fraudulentas no se diferenciaban claramente entre sí.

En centros de pago se encontró un tipo de transacciones que abarcaba casi todo el fraude y se

tenía una tasa de fraude del 25 % por lo que no se aplicó ningún modelo, al ser un caso atípico de fraude requeriría otro tipo de tratamiento y los resultados que se tendrían no serían comparables con los otros modelos.

Para comparar los resultados de los modelos en los rubros de recargas telefónicas y transacciones financieras, se construyeron modelos con todos los tipos de transacciones disponibles. Éstos últimos se aplicaron sobre los rubros escogidos y tuvieron un peor desempeño que los primeros. En recargas telefónicas el número de falsos negativos por cada verdadero positivo (nFP) se reduce entre un 35 % y 75 % y el indicador económico aumenta entre un 14 % y 109 %. Mientras que para transacciones financieras el nFP se reduce entre un 49 % y 80 %. Por lo tanto es mejor aislar tipos de transacciones, modelar el comportamiento de fraude en éstos y obtener varios modelos, que tener un solo modelo.

Las variables acumuladoras o históricas, las cuales revisan la historia con respecto a la agrupación de rubros creada, tienen gran poder predictivo, medido a través del KS (distancia de Kolmogorov-Smirnov). Mientras más historia se abarca, mayor es el KS. En general las variables más relevantes para los modelos aplicados fueron las acumuladoras de 12 meses y del día. En particular, para la base total las variables más relevantes fueron: la agrupación de rubro, las acumuladoras de 12 meses y del día. Para la base de recargas fueron: la agrupación del monto, las acumuladoras de 12 meses y 90 días. Mientras que para la base de trx. financieras fueron: las acumuladoras de 12 meses, la agrupación del rubro y las acumuladoras del día.

En las variables acumuladoras se podrían mejorar los resultados, pues mientras más historia se acumulaba, el KS aumentaba, luego se podrían considerar periodos mayores a 12 meses y se tendría un mayor poder predictivo. Otra forma de considerar variables acumuladoras, sería un acumulador variable, dependiendo de cada cliente, por ejemplo, considerar la historia desde la primera vez que transaccionó.

Los modelos propuestos para la detección de fraude fueron seis: support vector machines (SVM): una con kernel radial y otra con polinomial, artificial neural network (ANN), árboles de decisión: CART y C4.5, y regresión logística. En cuanto a tiempos de entrenamiento y test, los modelos SVM y ANN son considerablemente mayores, superiores a un día en el entrenamiento. Se compararon los modelos con diversas medidas de rendimiento, para capturar sus capacidades predictivas y de clasificación. En general SVM y ANN tuvieron mejores resultados.

Desde el punto de vista del negocio se desarrolló una metodología clara, ordenada y eficaz para la detección de fraude a nivel transaccional. En ésta se consideraron distintas maneras de construir los modelos, cuya elección depende de las necesidades y objetivos del negocio. Se mostraron varios tipos de modelos, donde los más complejos, como SVM y ANN, presentaron mejores resultados que modelos más comunes como la regresión logística y árboles de decisión. Sin embargo los primeros requieren de herramientas más sofisticadas para su implementación. Se recomienda al negocio la depuración de la base de datos, principalmente que las transacciones estén marcadas como normales o fraudulentas, para la implementación y evaluación de los modelos y su futuro mantenimiento y seguimiento.

Como conclusión general se puede decir que entrenar los modelos en bases con una menor proporción de casos fraudulentos entrega mejores resultados en la predicción global. Es mejor

aplicar diferentes modelos en cada rubro o tipo de transacción y tener varios modelos, que uno solo en general, es decir, hay una ganancia en dividir el problema. Modelos más complejos como SVM y ANN entregan mejores resultados que modelos más sencillos como regresión logística.

7.1. Trabajos Futuros

Con los resultado de los modelos, a posteriori se puede crear un modelo con pesos o costos para las distintas transacciones, ya que no todas las transacciones tienen la misma importancia en el negocio. También al momento de entrenar los modelos se puede dar distinta importancia a las transacciones normales y fraudulentas. Para esto se puede contruir una matriz de costos.

Es posible hacer un mayor análisis en los datos y en conjunto con los resultados obtenidos, buscar otros rubros o comercios donde se puedan encontrar patrones extraños de compra, como en recargas telefónica y enfocarse en éstos para detectar el fraude transaccional. Con esto se obtiene un árbol o segmentación y en cada hoja o segmento se aplica un modelo diferente.

A medida que se acumula más historia, se obtiene un mayor poder predictivo. En este trabajo se usaron variables acumuladoras con historia de hasta 12 meses, quizás al ver un periodo mayor se obtengan variables más relevantes para los modelos. También las variables acumuladoras se pueden considerar entre agrupaciones de rubros, no solo para la misma agrupación de rubro como fue hecho en esta memoria.

Para este estudio algunas variables propias de la transacción consideradas en los modelos tenían bajo KS, es recomendable eliminar algunas de estas variables con poco poder discriminante, tales como, el monto para la base total y la variable acumuladora de 30 días para la base de transacciones financieras.

En este trabajo sólo se usaron variables agrupadas en categorías, es decir, se usaron variables discretas. Para darle mayor riqueza a los modelos se pueden incluir variables continuas. Al agregar variables continuas se evita el problema de que muchas transacciones tengan la misma probabilidad de ser fraudulentas, lo que implica dificultades en medidas tales como el impacto económico.

Si se posee mayor poder de cálculo, se tiene la facultad de hacer un mejor entrenamiento de los modelos: probando una gama superior de parámetros y distintas proporciones de casos fraudulentos al entrenar. Algunos entrenamientos en la base con mayor números de transacciones (base total, 10 % de transacciones fraudulentas) tuvieron tiempos de más de un día, por lo que no se probaron bases con menos del 10 % de transacciones fraudulentas.

Por el tiempo disponible se usaron seis meses de transacciones. Para hacer modelos más estables se puede abarcar un periodo más amplio, por ejemplo doce meses. Es recomendable tener una base de datos consolidada con las transacciones de fraudulentas identificadas. Esto facilita la creación de modelos, su evaluación y posterior seguimiento y mantenimiento.

Esta memoria no contempló la implementación de modelos, sin embargo, en recargas telefónicas se obtuvo una evaluación económica positiva, luego se podría trabajar en la implementación de éstos.

Bibliografía

[1] Aleskerov, Emin y Rao Bharat: *A Neural Network Based Database stern for Credit Card Fraud Detection*. Signal Processing, páginas 220–226, 1997.

[2] Bentley, Peter J., Jungwon Kim, Gil H. Jung y Jong U. Choi: *Fuzzy Darwinian Detection of Credit Card Fraud*. En *14th Annual Falll Symposium of the Korean Information Processing Society*, 2000.

[3] Bhattacharyya, Siddhartha, Sanjeev Jha, Kurian Tharakunnel y J Christopher Westland: *Data mining for credit card fraud: A comparative study*. Decision Support Systems, 50(3):602–613, 2011.

[4] Bhusari, V. y S. Patil: *Study of Hidden Markov Model in Credit Card Fraudulent Detection*. International Journal of Computer Applications, 20(5):33–36, April 2011. Published by Foundation of Computer Science.

[5] Bolton, Richard J. y David J. H: *Unsupervised profiling methods for fraud detection*. En *Proc. Credit Scoring and Credit Control VII*, páginas 5–7, 2001.

[6] Breiman, L., J. Friedman, R. Olshen y C. Stone: *Classification and Regression Trees*. Wadsworth and Brooks, Monterey, CA, 1984.

[7] Bryson, A.E. y Y.C. Ho: *Applied Optimal Control*. Blaisdell book in the pure and applied sciences. Blaisdell Pub. Co., 1969.

[8] Chan, Philip K., Wei Fan, Andreas Prodromidis y Salvatore J. Stolfo: *Distributed Data Mining in Credit Card Fraud Detection*. IEEE Intelligent Systems, 14:67–74, 1999.

[9] Chen, Rong Chan, Shu Ting Luo, Xun Liang y V.C.S. Lee: *Personalized approach based on SVM and ANN for detecting credit card fraud.* En *IEEE International Conference on Neural Networks and Brain*, páginas 810–815, Octubre 2005, ISBN 0-7803-9422-4.

[10] Chiu, Chuang Cheng y Chieh Yuan Tsai: *A Web Services-Based Collaborative Scheme for Credit Card Fraud Detection.* En *EEE*, páginas 177–181. IEEE Computer Society, 2004, ISBN 0-7695-2073-1.

[11] Cortes, Corinna y Vladimir Vapnik: *Support-Vector Networks*. En *Machine Learning*, volumen 20, 1995.

[12] Diem, K.: *Documenta Geigy: Scientific tables*. J. R. Geigy, 1962.

[13] Elkan, Charles: *Magical thinking in data mining: lessons from CoIL challenge 2000.* En *KDD*, páginas 426–431, 2001.

[14] Estévez, Pablo A., Claudio M. Held y Claudio A. Perez: *Subscription fraud prevention in telecommunications using fuzzy rules and neural networks*. Expert Syst. Appl., 31(2):337–344, 2006.

[15] Fan, Wei: *Systematic data selection to mine concept-drifting data streams.* En Kim, Won, Ron Kohavi, Johannes Gehrke y William DuMouchel (editores): *KDD*, páginas 128–137. ACM, 2004, ISBN 1-58113-888-1.

[16] Fawcett, T.: *ROC Graphs: Notes and Practical Considerations for Data Mining Researchers*. Technical Report HPL-2003-4, HP Laboratories, 2003.

[17] Fayyad, Usama, Gregory Piatetsky-shapiro y Padhraic Smyth: *From Data Mining to Knowledge Discovery in Databases*. AI Magazine, 17:37–54, 1996.

[18] Hastie, T., R. Tibshirani y J. Friedman: *The elements of statistical learning: Data mining, inference, and prediction*. Springer, New York, 2001.

[19] Haykin, Simon: *Neural Networks: A Comprehensive Foundation*. Prentice Hall, 1999.

[20] Hilas, Constantinos S. y John N. Sahalos: *User Profiling for Fraud Detectionin Telecommunication Networks*. En *In Fifth international conference on technology and automation*, páginas 382–387, 2005.

[21] Hilbe, J. M.: *Logistic Regression Models*. Chapman & Hall/ Crc: Texts in Statistical Science Series. Chapman & Hall/CRC, 2009.

[22] Hollmén, Jaakko y Volker Tresp: *Call-Based Fraud Detection in Mobile Communication Networks Using a Hierarchical Regime-Switching Model*. En Kearns, Michael J., Sara A. Solla y David A. Cohn (editores): *NIPS*, páginas 889–895. The MIT Press, 1998, ISBN 0-262-11245-0.

[23] Hosmer, David W. y Stanley Lemeshow: *Applied logistic regression (Wiley Series in probability and statistics)*. Wiley-Interscience Publication, spanish2 edición, 2000, ISBN 0471356328.

[24] Hunt, E. B.: *Concept learning: an information processing problem*. Wiley, New York, 1962.

[25] Hyafil, Laurent y R. L. Rivest: *Constructing Optimal Binary Decision Trees is NP-complete*. Information Processing Letters, 5(1):15–17, 1976.

[26] Kim, Min Jung y Taek Soo Kim: *A Neural Classifier with Fraud Density Map for Effective Credit Card Fraud Detection*. En *Proceedings of the Third International Conference on Intelligent Data Engineering and Automated Learning*, IDEAL '02, páginas 378–383, London, UK, UK, 2002. Springer-Verlag, ISBN 3-540-44025-9.

[27] Kohavi, Ron y Ross Quinlan: *Decision Tree Discovery*. En *IN HANDBOOK OF DATA MINING AND KNOWLEDGE DISCOVERY*, páginas 267–276. University Press, 1999.

[28] Kokkinaki, A. I.: *On Atypical Database Transactions: Identification of Probable Frauds using Machine Learning for User Profiling*. Knowledge and Data Exchange, IEEE Workshop on, 0:107, 1997.

[29] Kundu, Amlan, Shamik Sural y Arun K. Majumdar: *Two-Stage Credit Card Fraud Detection Using Sequence Alignment.* En Bagchi, Aditya y Vijayalakshmi Atluri (editores): *ICISS*, volumen 4332 de *Lecture Notes in Computer Science*, páginas 260–275. Springer, 2006, ISBN 3-540-68962-1.

[30] Lim, Tjen S., Wei Y. Loh y Yu S. Shih: *A Comparison of Prediction Accuracy, Complexity, and Training Time of Thirty-Three Old and New Classification Algorithms*. Machine Learning, 40(3):203–228, 2000.

[31] Maes, Sam, Karl Tuyls y Bram Vanschoenwinkel: *Machines Learning Techniques for Fraud Detection*. Master thesis, Vrije Universiteit Brussel, 2000.

[32] Maes, Sam, Karl Tuyls, Bram Vanschoenwinkel y Bernard Manderick: *Credit Card Fraud Detection Using Bayesian and Neural Networks*. En *First International NAISO Congress on Neuro Fuzzy Technologies*, 2002.

[33] Morgan, J. N. y J. A. Sonquist: *Problems in the analysis of survey data, and a proposal*. Journal of the American Statistical Association, 58:415–434, 1963.

[34] Ngai, E W T, Yong Hu, Y H Wong, Yijun Chen y Xin Sun: *The application of data mining techniques in financial fraud detection: A classification framework and an academic review of literature*. Decision Support Systems, 50(3):559–569, 2011.

[35] Panigrahi, Suvasini, Amlan Kundu, Shamik Sural y Arun K. Majumdar: *Credit card fraud detection: A fusion approach using Dempster-Shafer theory and Bayesian learning.* Information Fusion, 10(4):354–363, 2009.

[36] Phua, Clifton, Damminda Alahakoon y Vincent Lee: *Minority report in fraud detection: classification of skewed data*. ACM SIGKDD Explorations Newsletter, 6(1):50–59, 2004.

[37] Phua, Clifton, Vincent Lee, Kate Smith-Miles y Ross Gayler: *A Comprehensive Survey of Data Mining-based Fraud Detection Research*. 2005.

[38] Quinlan, J. R.: *Discovering Rules by Induction from Large Collections of Examples*. En Michie, D. (editor): *Expert Systems in the Micro-Electronic Age*. Edinburgh University Press, Edinburgh, 1979.

[39] Quinlan, J. R. y R. L. Rivest: *Inferring decision trees using the minimum description length principle*. Inf. Comput., 80(3):227–248, Marzo 1989, ISSN 0890-5401.

[40] Rosenblatt, F.: *On the convergence of reinforcement procedures in simple perceptrons*. Report No VG-1196-G4. Cornell Aeronautical Laboratory, Inc., Buffalo, New York, Febrero 1960.

[41] Sentz, Kari y Scott Ferson: *Combination of evidence in Dempster-Shafer theory*. Informe técnico, 2002.

[42] Srivastava, Abhinav, Amlan Kundu, Shamik Sural y Arun Majumdar: *Credit Card Fraud Detection Using Hidden Markov Model.* IEEE Trans. Dependable Sec. Comput., 5(1):37–48, 2008.

[43] Umayaparvathi, V. y K. Iyakutti: *A Fraud Detection Approach in Telecommunication using Cluster GA*. International Journal of Computer Trends and Technology, páginas 40–45, 2011.

[44] Venables, William N. y Brian D. Ripley: *Modern Applied Statistics with S. Fourth Edition*. Springer, 2002. ISBN 0-387-95457-0.

[45] Wheeler, Richard y J. Stuart Aitken: *Multiple algorithms for fraud detection.* Knowl.-Based Syst., 13(2-3):93–99, 2000.

Apéndice A

Medidas de rendimiento

Para definir las medidas se usan los valores definidos en 2.3:

- Verdadero positivo (VP): la verdadera clase es positiva y se predice positiva.
- Falso negativo (FN): la verdadera clase es positiva y se predice negativa.
- Verdadero negativo (VN): la verdadera clase es negativa y se predice negativa.
- Falso positivo (FP): la verdadera clase es negativa y se predice positiva.

A.1. Sensibilidad y especificidad

La sensibilidad es la precisión en la clase positiva y se calcula como:

$$\text{Sensibilidad} = \frac{VP}{VP+FN} \tag{A.1}$$

Toma valores entre 0 y 1. El valor 0 indica ningún caso positivo es predicho correctamente y el valor 1 que todos los casos positivos se predicen correctamente.

La especificidad es la precisión en la clase negativa y se calcula como:

$$\text{Especificidad} = \frac{VN}{VN+FP} \tag{A.2}$$

Toma valores entre 0 y 1. El valor 0 indica ningún caso negativo es predicho correctamente y el valor 1 que todos los casos negativos se predicen correctamente.

A.2. Precisión global

La precisión global es el porcentaje de casos correctamente clasificados. Se calcula como:

$$\text{Prec. global} = \frac{(TP+TN)}{(TP+FP+TN+FN)} \tag{A.3}$$

Toma valores entre 0 y 1. El valor 0 indica ningún caso predicho correctamente y el valor 1 que todos los casos se predicen correctamente, es decir, una predicción perfecta.

A.3. nFP

El nFP es el número de falsos negativos por cada verdadero positivo y se calcula como:

$$nFP = \frac{FN}{VP} \tag{A.4}$$

Toma valores no negativos. Mientras más bajo el valor, mejor clasifica el modelo.

A.4. MCC

Matthews correlation coefficient (MCC) es usado como medida de la calidad de clasificaciones binarias. Toma en cuenta los verdaderos y falsos positivos y negativos y generalmente es considerado una medida balanceada que puede ser usada incluso si las clases son de tamaños muy diferentes. Se calcula como:

$$MCC = \frac{VP \times VN - FP \times FN}{\sqrt{(VP+FP)(VP+FN)(VN+FP)(VN+FN)}} \tag{A.5}$$

Toma valores entre -1 y 1. Un valor de -1 indica desacuerdo total entre lo predicho y observado, 0 indica que la predicción no es mejor que elección aleatoria y 1 representa una predicción perfecta.

A.5. KS

El KS (distancia de Kolmogorov-Smirnov) de una variable se define como la máxima diferencia de las distribuciones de la variable en las categorías de una variable respuesta, así el KS

indica cuanto discrimina una variable a otra. El cálculo del KS se muestra en diversas tablas en esta memoria. El KS toma valores entre 0 y 1, aunque también es común representarlo en porcentajes con valores entre 0 % y 100 %. El valor 0 (0 %) indica que la variable respuesta no discrimina y el valor 1 (100 %) representa discriminación perfecta.

A.6. AUC

AUC (*area under the ROC curve*, área bajo la curva ROC), como su nombre indica, mide el área bajo la curva ROC. La curva ROC (*Receiver Operating Characteristics*) es una técnica para visualizar, organizar y seleccionar clasificadores basado en su rendimiento. El gráfico de una curva ROC es bi-dimensional, en donde la sensibilidad se grafica en el eje Y y 1-Especificidad en el eje X. Una curva ROC describe el *trade-off* entre los beneficios (VP) y los costos (FP). En el apéndice C se pueden ver varios ejemplos de curvas ROC. El punto (0,1) representa la clasificación perfecta. La diagonal y=x representa la estrategia de adivinar al azar una clase. Luego, informalmente, un buen clasificador se encuentra en la región superior a la diagonal y con una tendencia al punto (0,1). Esto se puede medir mediante AUC. Éste toma valores entre 0 y 1, sin embargo como la diagonal y=x representa predicción al azar y tiene un área de 0,5, ningún clasificador realista debería tener un AUC menor a 0,5.

Apéndice B

Detección de fraude a nivel cliente

El modelo a nivel cliente es un modelo previo al transaccional, el cual es el objetivo principal de esta memoria. El modelo a nivel cliente se hizo, pues los datos a nivel transaccional en un principio no estaban disponibles y necesitaban un gran tratamiento de datos, en cambio a nivel cliente los datos son más fáciles de manejar. Este modelo no pretende ser aplicable, más bien sirve para ver el fraude a un nivel macro, por esta razón y por que no es el objetivo principal de la memoria, no se hizo un gran ajuste a los modelos.

B.1. Datos a nivel cliente

Con la bases de fraude se construyeron bases mensuales a nivel cliente, es decir, clientes que tienen transacciones fraudulentas en un determinado mes. Se consideró un periodo de ocho meses. Los archivos contienen el número de cuenta del cliente, el monto total fraudulento, fecha (año mes) y n° de casos fraudulentos. La estructura de estos archivos se muestra en la tabla B.1.

Los datos de los clientes se encuentran en una base identificada por el n° de cuenta del cliente, la cual se nombra en adelante como "base de clientes". Esta base se genera mensualmente y contiene datos tales como la edad del cliente, el cupo de la tarjeta, la deuda, entre otros. Como se desea detectar o predecir si un cliente se verá afectado por fraude, la base de clientes de un mes dado se le agrega los datos de fraude del mes siguiente.

Cuenta	Monto_Fraude	Fecha	N_Fraude
1	53.499	Mes 1	2
2	4.682	Mes 1	1
3	118.992	Mes 1	9

Tabla B.1: Estructura base de fraude a nivel cliente.

N°	Fecha	N° Fraude	N° Fraude (*)
1	Mes 1	138	138
2	Mes 2	129	105
3	Mes 3	154	137
4	Mes 4	152	123
5	Mes 5	227	199
6	Mes 6	219	183
7	Mes 7	232	199
8	Mes 8	201	173
(*): N° de cuenta único.			

Tabla B.2: Número de clientes con fraude en el periodo considerado. (*): N° cuenta único.

En la tabla B.2 se encuentra el n° de clientes con fraude del periodo considerado. La cantidad total de clientes es grande, aproximadamente dos millones, luego para construir un modelo se toman muestras. Por el contrario la cantidad de clientes que sufren fraude es pequeña, luego para construir un modelo correctamente se construyen bases equilibradas, es decir, con igual número de clientes con fraude y sin fraude. Además se consideran clientes únicos, es decir, un cliente con fraude un mes ya no se considera en los siguientes aunque haya sufrido fraude, esto es para evitar darle más importancia a estos casos repetidos en el modelo.

Para construir los modelos se sacaron tres muestras equilibradas y con estas se construyeron tres muestras consolidadas de los ocho meses, en adelante denotadas como M1, M2 y M3, respectivamente. En la tabla B.3 se encuentra una lista con las variables a considerar en los modelos.

Las primeras 12 variables son propias del cliente y su cuenta, las variables 13 a la 24 son calculadas a partir de datos históricos y la variable 25 es la que indica si un cliente tuvo fraude. Para obtener mejores resultados en los modelos las variables de escala o categóricas con varias categorías, se agruparon en variables con menos categorías, de dos a cinco categorías cada una, excepto por la edad que se dejó con 10 categorías. Las nuevas categorías se crearon analizando las distribuciones de casos fraudulentos en las categorías previas, y se ordenaron las nuevas categorías según la tasa de fraude. A modo de ejemplo en la tabla B.4 se encuentran agrupadas las categorías de la variable “Sop_compras”.

La nueva variable, llamada “Sop_compras_Ag”, posee tres categorías, la primera categoría corresponde al valor 0, la segunda categoría corresponde a los valores entre 1 y 4 y la tercera corresponde a los valores entre 5 y 10.

N°	Variable	N° Descripción
1	Edad	Edad en años
2	Cod_act	Código de actividad
3	Ant_cuen	Antigüedad de la cuenta en años
4	Cupo	Cupo
5	Saldo	Saldo
6	Disponible	Disponible
7	NSE	Nivel socioeconómico
8	Sexo	Sexo
9	Region	Región geográfica
10	Tarj_cred	Tipo de tarjeta de crédito
11	Sald_cupo	Relación saldo cupo
12	Pago_deud	Relación pago deuda
13	Sop_compras	Suma del soporte de compras en últimos 12 meses
14	Marca_ultmov	Marca si se hizo movimiento en mes anterior
15	Marca_mov12M	Marca si se hizo movimiento en últimos 12 meses
16	Marca_avan12M	Marca si se usó avance en los últimos 12 meses
17	Marca_avan24M	Marca si se usó avance en los últimos 24 meses
18	Marca_CC12M	Marca si se usó CC en los últimos 12 meses
19	Marca_CC24M	Marca si se usó CC en los últimos 24 meses
20	Marca_CC36M	Marca si se usó CC en los últimos 36 meses
21	Marca_CC	Marca si se usó CC alguna vez
22	Rent_tarj	Rentabilidad tarjeta
23	Rent_tiend	Rentabilidad tienda
24	Score	Score de crédito
25	flag_fraude	Flag si el cliente tuvo fraude

Tabla B.3: Variables a considerar en los modelos.

	flag_impug			
Sop_compras	No	Si	Total	Tasa
0	173	35	208	17 %
1	12	10	22	45 %
2	50	22	72	31 %
3	191	119	310	38 %
4	273	232	505	46 %
5	234	218	452	48 %
6	154	223	377	59 %
7	97	172	269	64 %
8	48	167	215	78 %
9	23	52	75	69 %
10	2	7	9	78 %
Total	1.257	1.257	2.514	100 %

Tabla B.4: Agrupación de variable "Sop_compras" en tres categorías.

B.2. Entrenamiento

Los algoritmos que se proponen para resolver este problema son de tipo supervisado (o predictivo). El conjunto de datos de entrenamiento son las muestras consolidadas de las bases de cliente del periodo seleccionado y los datos test serán los meses siguientes. Las variables input son las variables ya agrupadas correspondiente a las mencionadas en la tabla B.3, excepto el flag de fraude que es la variable objetivo.

En la tabla B.5 se muestra el KS obtenido de cada variable agrupada que entra en los modelos.

En este caso el KS es de las variables agrupadas con el flag de fraude como variable respuesta, luego mide el nivel de separación entre las distribuciones acumuladas de clientes que tuvieron fraude y los que no. El KS toma valores entre 0 % y 100 %, el valor 0 % significa que la variable no puede discriminar a la variable respuesta y mientras más cercano al 100 % existe una mayor discriminación. El sufijo "Ag" en las variables de la tabla B.5 se utiliza para indicar que esa variable está agrupada en categorías (como se mencionó anteriormente), si no tiene sufijo quiere decir que la variable es tal como venía en las bases.

Con las variables ya agrupadas, las muestras consolidadas se ingresan en un software con el que se crearon seis modelos para cada muestra:

- SVM
    1. *Kernel* polinomial
    2. *Kernel* radial
- Regresión logística binaria

N°	Variable	KS_M1	KS_M2	KS_M3
1	Edad_Ag	7%	8%	7%
2	Cod_act_Ag	9%	9%	9%
3	Ant_cue_Ag	8%	8%	10%
4	Cupo_Ag	19%	18%	17%
5	Saldo_Ag	19%	17%	18%
6	Disponible_Ag	15%	13%	11%
7	NSE_Ag	9%	7%	6%
8	Sexo_Ag	0%	4%	4%
9	Region_Ag	6%	7%	5%
10	Tarj_cred	7%	8%	7%
11	Saldo_cupo_Ag	19%	20%	21%
12	Pago_deud_Ag	11%	11%	10%
13	Sop_compras_Ag	21%	22%	20%
14	Marca_ultmov	19%	20%	21%
15	Marca_mov12M	16%	16%	16%
16	Marca_avan12M	8%	7%	6%
17	Marca_avan24M	10%	7%	7%
18	Marca_CC12M	5%	5%	4%
19	Marca_CC24M	7%	7%	5%
20	Marca_CC36M	8%	8%	5%
21	Marca_CC	10%	8%	6%
22	Rent_tarj_Ag	15%	14%	14%
23	Rent_tiend_Ag	11%	7%	8%
24	Score_Ag	22%	21%	19%

Tabla B.5: KS de las variables que entran en los modelos.

N°	Modelo	Precisión_M1	Precisión_M2	Precisión_M3
1	SVM polinomial	0,998	0,999	0,997
2	SVM radial	0,995	0,997	0,995
3	C4.5	0,741	0,749	0,753
4	Logística	0,708	0,695	0,707
5	Red neuronal	0,686	0,704	0,699
6	CART	0,689	0,684	0,689

Tabla B.6: Precisión global en el entrenamiento de los modelos.

- Árboles de decisión
    1. C4.5
    2. CART
- ANN

En la tabla B.6 se muestra la precisión global obtenida para los modelos generados en las tres muestras consolidadas. Se encuentran ordenados de mayor a menor según la precisión global de la muestra M1.

La precisión global se mantiene estable en las tres muestras para todos los modelos. Ésta es superior en las SVMs siendo cercana a 1, pero un problema de los SVMs es que puede haber *overfitting*, es decir, el modelo se sobre ajusta a los datos de entrenamiento y no clasifica bien otros datos que no entraron en la construcción del modelo. Esto último se puede ver con los datos test, éstos corresponden a la base de clientes de los cinco meses siguientes, es decir, desde el Mes 9 al 13, a los que se les agregó el flag de fraude.

B.3. Resultados

Los modelos que se aplicarán en los datos test son: el SVM polinomial, el SVM Base radial y el C4.5, pues obtuvieron los mejores resultados; la regresión logística, pues es bastante utilizada y conocida en el negocio; y por último la red neuronal, pues es muy utilizada en las publicaciones sobre detección de fraude. Se deja a fuera el modelo CART, por tener las precisiones más bajas. Sólo se consideran los modelos entrenados con la muestra M1, pues se mostró estabilidad en la precisión global para las diferentes muestras.

En las tablas B.7 y B.8 se aprecian los resultados de precisión global y KS de los modelos en las bases de clientes del Mes 8 al Mes 12, respectivamente. En este caso el KS es del valor predicho vs el valor observado.

La precisión de los modelos se mantiene estable en los conjuntos test, siendo mayor en los modelos de regresión logística y el C4.5 y más baja para la red neuronal. Por otro lado el KS

	Precisión Global					
Modelo	Mes 8	Mes 9	Mes 10	Mes 11	Mes 12	Prom. Mes 9-12
Logística	0,692	0,685	0,690	0,687	0,686	0,687
SVM radial	0,593	0,584	0,585	0,593	0,590	0,588
SVM polinomial	0,611	0,603	0,604	0,615	0,612	0,609
C4.5	0,664	0,664	0,665	0,635	0,638	0,650
Red neuronal	0,464	0,465	0,471	0,440	0,440	0,454

Tabla B.7: Precisión global modelos.

	KS					
Modelo	Mes 8	Mes 9	Mes 10	Mes 11	Mes 12	Prom. Mes 9-12
Logística	31 %	31 %	27 %	27 %	32 %	29 %
SVM radial	48 %	22 %	15 %	17 %	18 %	18 %
SVM polinomial	46 %	18 %	13 %	13 %	16 %	15 %
C4.5	28 %	25 %	23 %	20 %	22 %	23 %
Red neuronal	28 %	30 %	24 %	25 %	27 %	26 %

Tabla B.8: KS modelos.

se mantiene estable en todos los modelos excepto las SVMs, donde se observa un gran descenso del KS en los meses tests, que no fueron usados para construir el modelo. Esto último muestra el *overfitting* de las SMVs, ya que el KS obtenido en el mes de entrenamiento es muy alto, pero en los datos test son los más bajo de todos. El KS más alto se obtiene en la regresión logística y la red neuronal.

En la figura B.1 se muestran las curvas ROC para la regresión logística, la SVM polinomial, la SVM radial, la red neuronal y el C4.5 para el Mes 9.

Analizando los resultados de los dos indicadores, precisión global y KS, y las curvas ROC, podemos decir que el modelo que muestra mejores resultados es la regresión logística. Por esta razón se tratará de mejorar la performance de este tipo de modelo.

Como el n° de variables introducidas es grande, algunas no aportan un peso importante a la logística. Analizando el peso de cada variable y su KS, con expertos del negocio se decidió eliminar y fusionar algunas variables y fusionar categorías. Finalmente las variables a entran al modelo son 15. Éstas se encuentran en la tabla B.9 con el número de categorías. La única nueva variable es *Compras_12M* que es una combinación de las variables *Sop_compras_Ag* y *Marca_mov12M*.

En la tabla B.10 se muestran los resultados en cuanto a precisión global y KS de las nuevas logísticas.

Si se comparan estos resultados con los obtenidos en la tabla B.8 en la logística anterior se puede ver que en promedio la precisión baja 4 % y el KS se mantiene, pero esto se compensa

N°	Variable	N° categorías
1	Edad_Ag	10
2	Cod_act_Ag	3
3	Ant_cue_Ag	2
4	Cupo_Ag	4
5	Saldo_Ag	3
6	Disponible_Ag	2
7	Region_Ag	3
8	Tarj_cred	3
9	Sald_cupo_Ag	3
10	Pago_deud_Ag	4
11	Compras_12M	3
12	Marca_ultmov	2
13	Marca_avan24M	2
14	Marca_CC12M	2
15	Score_Ag	4

Tabla B.9: Variables definitivas a considerar en los modelos y n° de categorías.

	Mes 8		Mes 9		Mes 10		Mes 11		Mes 12	
Modelo	KS	Prec.	KS	Prec.	KS	Prec.	KS	Prec.	KS	Prec.
Logística M1	32 %	0,666	29 %	0,659	28 %	0,665	27 %	0,637	30 %	0,636
Logística M2	28 %	0,665	31 %	0,655	29 %	0,657	28 %	0,621	29 %	0,621

Tabla B.10: Precisión global logísticas.

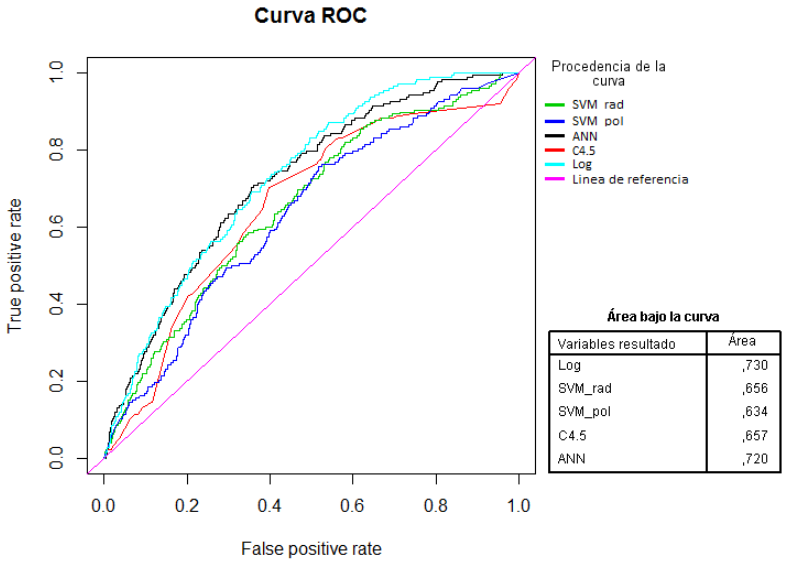

Figura B.1: Curvas ROC Mes 9.

con el menor número de variables, pues una de las ventajas de este modelo es que otorga pesos o importancia a las variables, lo que ayuda a explicar la predicción.

Para comparar las nuevas logísticas se utilizaron otros dos modelos construidos con las 15 variables sobre la muestra M1: SVM y C4.5.

En la figura B.2 se muestra la curva ROC para los modelos SVM, C4.5 y la logística construida con M1 para el Mes 9.

Luego con los resultados de la figura B.2 se puede decir que la logística es un mejor clasificador que la SVM y C4.5.

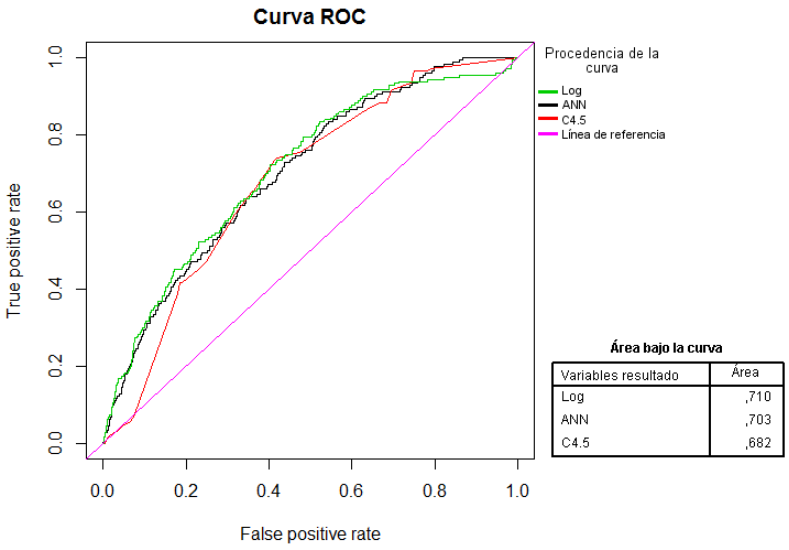

Figura B.2: Curvas ROC Mes 9, modelos con 15 variables.

Apéndice C

Curvas ROC

A continuación se encuentran las curvas ROC de los modelos a nivel transaccional para las muestras E50, E25 y E10, para las bases total, recargas telefónicas y transacciones financieras, vistos en el capítulo 6. La medida AUC (área bajo la curva) se calcula a partir de estas curvas ROC.

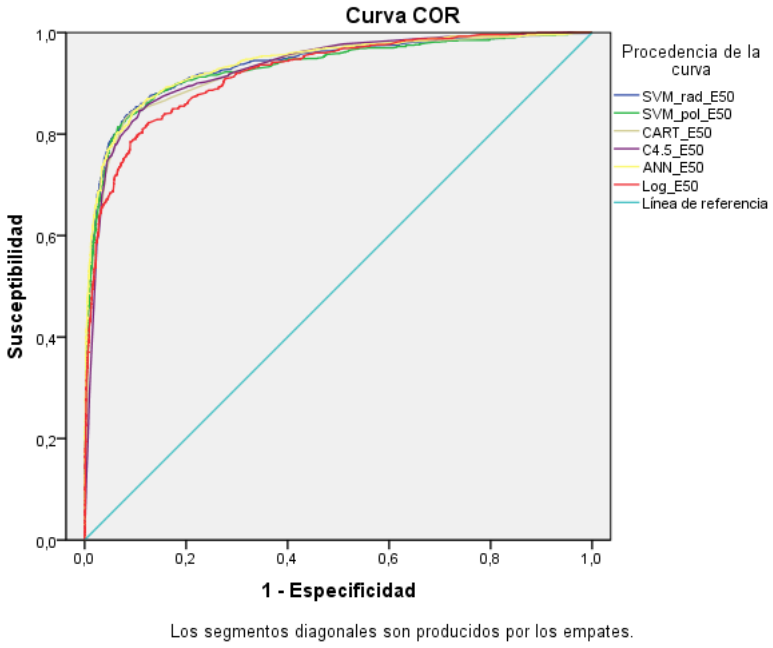

Figura C.1: Curvas ROC para los modelos entrenados en la base E50, base total.

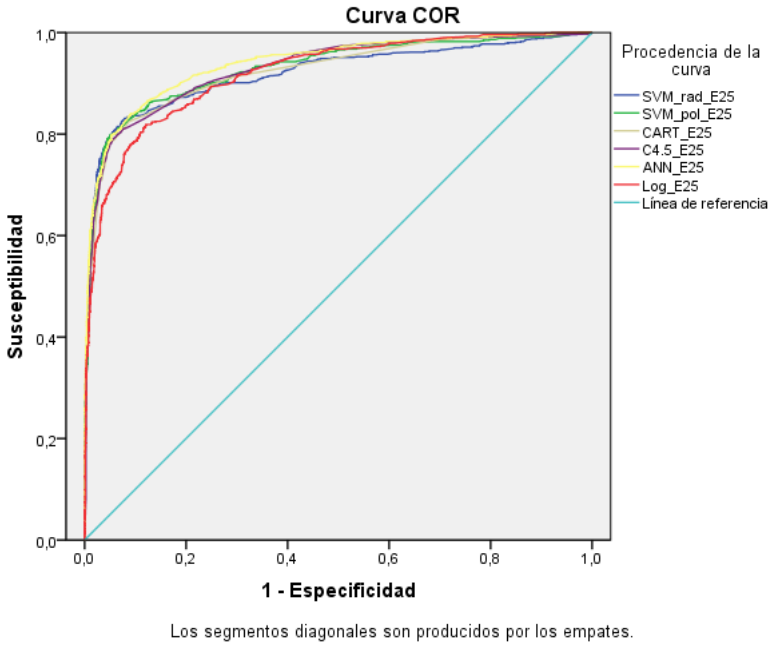

Figura C.2: Curvas ROC para los modelos entrenados en la base E25, base total.

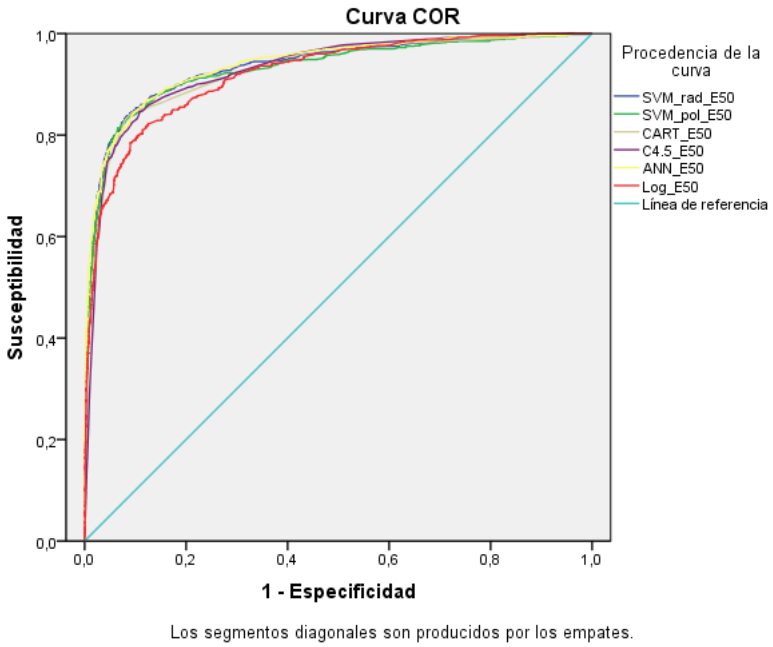

Figura C.3: Curvas ROC para los modelos entrenados en la base E10, base total.

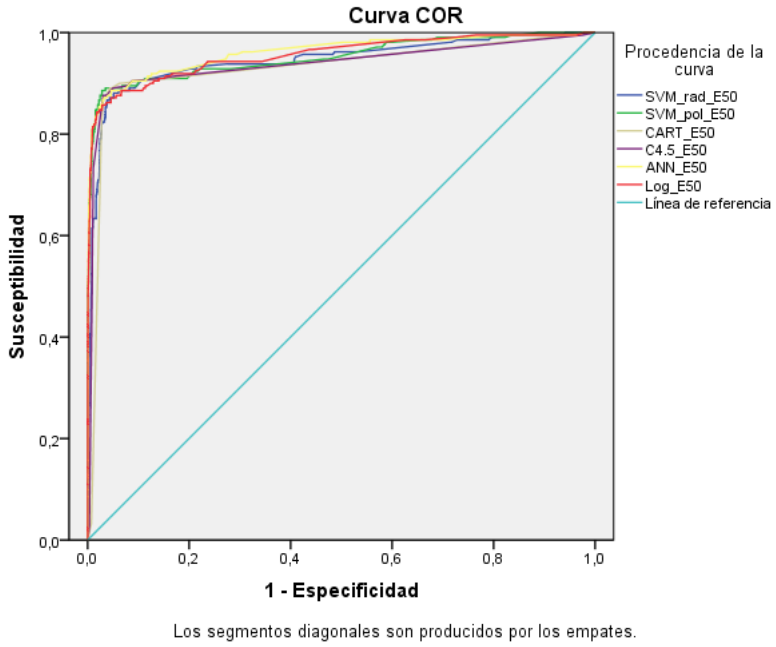

Figura C.4: Curvas ROC para los modelos entrenados en la base E50, base recargas.

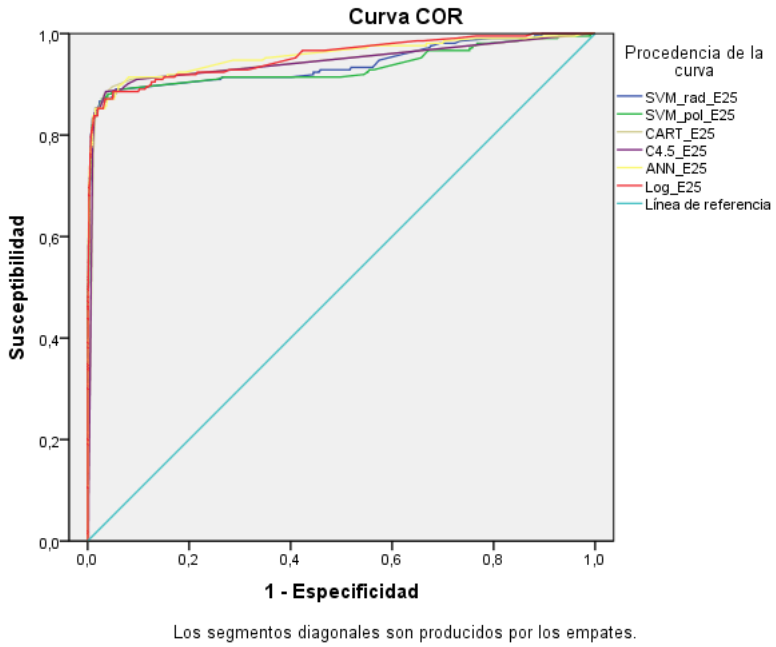

Figura C.5: Curvas ROC para los modelos entrenados en la base E25, base recargas.

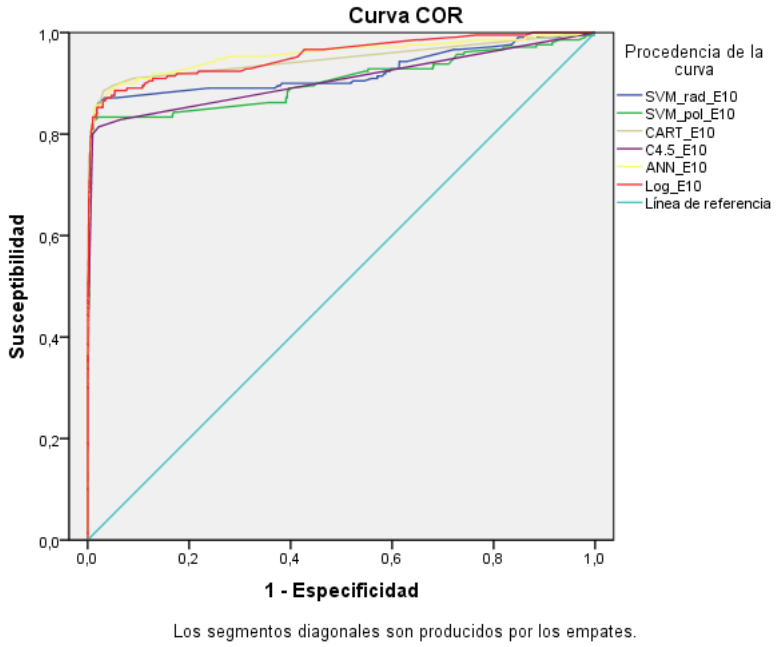

Figura C.6: Curvas ROC para los modelos entrenados en la base E10, base recargas.

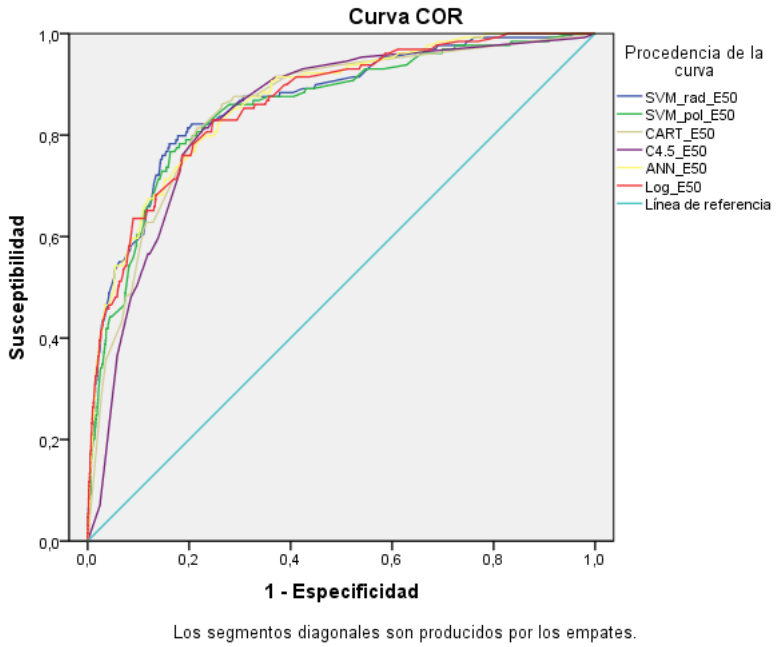

Figura C.7: Curvas ROC para los modelos entrenados en la base E50, base trx. financieras.

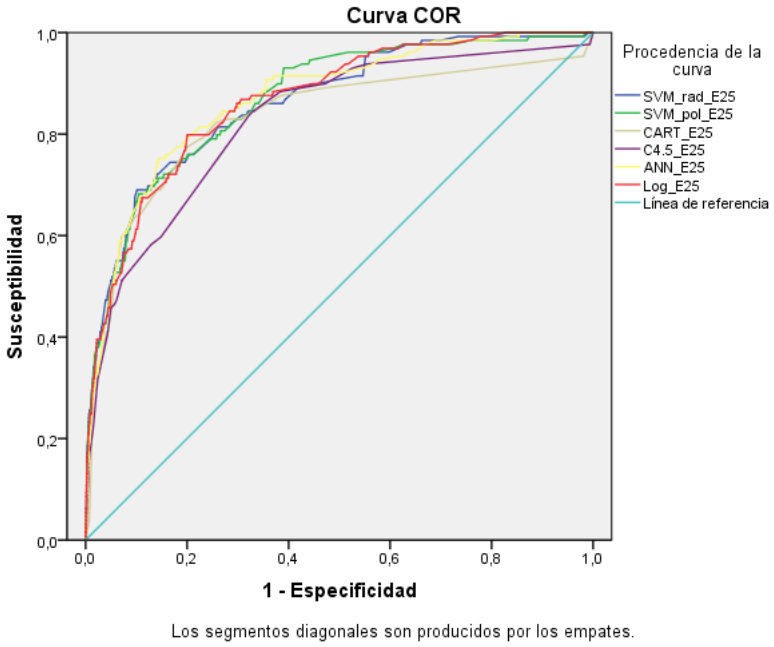

Figura C.8: Curvas ROC para los modelos entrenados en la base E25, base trx. financieras.

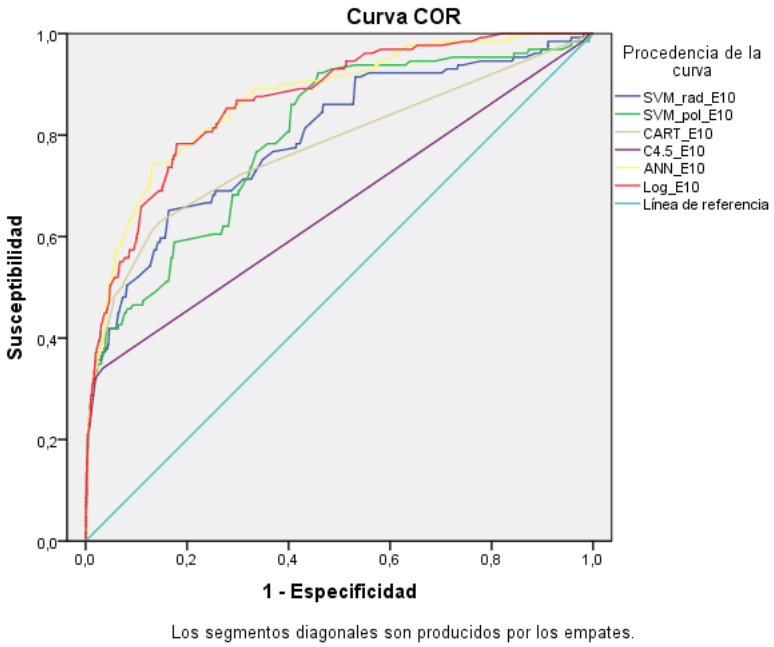

Figura C.9: Curvas ROC para los modelos entrenados en la base E10, base trx. financieras.